

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Étude, conception et implémentation d'un traitement de texte pour cadres supérieurs

Jeanjot, Bruno

Award date:
1985

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur

Institut d'informatique

Année académique 1984 - 1985

ETUDE, CONCEPTION ET IMPLEMENTATION

D'UN TRAITEMENT DE TEXTE

POUR CADRES SUPERIEURS

Bruno JEANJOT

Mémoire présenté en vue de l'obtention
du grade de Licencié et Maître
en Informatique

Nous remercions tout d'abord Monsieur Roland Lesuisse qui a accepté la direction de ce mémoire. Nous laissant une grande liberté dans le choix des décisions à prendre, il a guidé ce travail par ses remarques critiques et ses conseils judicieux tant au niveau conception et implémentation qu'au moment de la rédaction.

Nous tenons aussi à remercier tous ceux qui de près ou de loin ont contribué à la réalisation de ce projet.

TABLE DES MATIERES

<u>INTRODUCTION</u>	1
 <u>I. ETUDE GENERALE</u>	 3
1. QUESTIONS DE VOCABULAIRE	4
1.1. La terminologie	4
1.2. Notions de base	5
2. ETUDE DU PROCESSUS DE CREATION D'UN DOCUMENT	6
2.1. Constitution matérielle d'un système de de traitement de texte	6
2.2. Modèle sous-jacent à la production d'un document	7
3. LE TRAITEMENT DE TEXTE POUR CADRE SUPERIEUR	9
3.1. Définition du système	9
3.2. Spécificité du traitement de texte pour cadre supérieur	10
3.2.1. POURQUOI UN NOUVEAU TRAITEMENT DE TEXTE ?	10
3.2.2. TRAITEMENT POUR CADRE SUPERIEUR	11
 <u>II. MODELE CONCEPTUEL D'UN DOCUMENT</u>	 13
1. MODELE CONCEPTUEL DES DONNEES	14
1.1. Définition d'un document	14
1.2. Le modèle conceptuel	14
1.2.1. LES STRUCTURES D'UN DOCUMENT	15
1.2.2. DEFINITION DES ELEMENTS DE CES STRUCTURES	16
1.3. Les différents types de document	22
1.3.1. LETTRE	22
1.3.2. LETTRE STANDARD	22
1.3.3. FORMULAIRE	22
1.3.4. QUELCONQUE	22
2. LES FONCTIONS DU TRAITEMENT DE TEXTE	23
2.1. Fonctions destinées à faciliter l'opération de première frappe	23
2.2. Fonctions destinées à faciliter la correction d'un document	26
2.3. Fonctions destinées à faciliter l'opération de mise en forme	27
2.4. Fonctions destinées à faciliter la manipulation de documents	29
2.5. Fonction destinée à faciliter l'impression d'un document	30
2.6. Fonctions destinées à se déplacer dans le texte	30

<u>III. CONCEPTION DE L'INTERFACE</u>	33
1. ANALYSE DE DIFFERENTS TYPES D'INTERFACES	34
1.1. Wordstar	34
1.2. Macwrite	35
1.3. Professional editor	36
2. DESCRIPTION DE L'INTERFACE	37
2.1. But de l'interface	37
2.2. Description	37
2.2.1. LES ELEMENTS LOGICIELS	37
2.2.2. LES ELEMENTS MATERIELS	44
<u>IV. IMPLEMENTATION DU TRAITEMENT DE TEXTE</u>	52
1. DESCRIPTION DES FICHIERS ET UTILISATION	53
1.1. Le fichier "document"	53
1.1.1. DESCRIPTION	53
1.1.2. STRUCTURE DE MANIPULATION	54
1.2. Le fichier "données document"	57
1.3. Le fichier "tabulation"	58
1.4. Le fichier "liste d'adress"	58
1.5. Le fichier "dictionnaire"	60
2. L'ARCHITECTURE DU SYSTEME	61
2.1. Introduction	61
2.2. Analyse de la découpe modulaire	62
3. SPECIFICATION DES DIFFERENTS MODULES	66
3.1. Spécification des modules de niveau 0	67
3.2. Spécification des modules de niveau 1	69
3.3. Spécification des modules de niveau 2	75
3.3.1. MODULES DU MANAGER	75
3.3.2. MODULES DU GESTIONNAIRE D'ECRAN	82
3.3.3. MODULES DE L'EDITEUR	90
3.3.4. MODULES DU FORMATEUR	98

<u>V. MANUEL D'UTILISATION</u>	102
1. QUE FAUT-IL SAVOIR AVANT L'UTILISATION DU TRAITEMENT DE TEXTE	103
2. LE DEMARRAGE DU PROGRAMME	103
3. QUE SUFFIT-IL DE SAVOIR PENDANT L'UTILISATION DU TRAITEMENT DE TEXTE	103
3.1. Connaissance de l'écran	103
3.2. Connaissance du clavier	104
4. EXPLICATION DES TOUCHES ET COMMANDES	105
5. UN EXEMPLE D'UTILISATION	108

CONCLUSIONS

LEXIQUE

BIBLIOGRAPHIE

INTRODUCTION

Dans son ouvrage consacré à la bureautique, J. Martineau écrit : "Une simple correspondance revient actuellement à plusieurs dizaines de francs et ce coût s'est accru de 26 % en deux ans. Si cet état de fait se prolonge, dans quelques temps, la simple lettre rédigée, frappée et expédiée par un service deviendra, de par son prix de revient, un produit de luxe" (Martineau, 1, p.23).

Si l'on s'en tient à la correspondance expédiée par un cadre supérieur, on comprend aisément que son coût est déjà aujourd'hui une source de frais insoupçonnée.

Les traitements de texte - qui prennent en charge toute la présentation, toute la mise en page d'un texte pour que l'utilisateur ne se soucie plus que de son contenu - permettent de comprimer les coûts de production des documents.

Malheureusement, leur utilisation nécessite trop souvent un apprentissage important (un cours d'initiation à Wordstar dure 30 heures) ainsi qu'une utilisation régulière.

Le but de ce mémoire est d'offrir aux cadres supérieurs un outil très facile à utiliser pour simplifier la réalisation de leurs travaux écrits courants, tels que la production de lettres, de petits rapports, de mémos, ...

Le travail comporte cinq parties. Dans le premier chapitre, nous présentons, brièvement, le problème de la création de documents ainsi que le modèle sous-jacent, nous définissons un cadre de référence pour la terminologie et enfin, nous analysons le cas particulier d'un traitement de texte pour cadre supérieur.

Au chapitre deux, nous définissons le modèle conceptuel : définition des structures de représentation d'un document d'une part, et définition de la fonctionnalité de l'outil que désire un cadre supérieur d'autre part.

La troisième partie aborde la conception de l'interface. Après une brève critique d'interfaces existants, nous présentons les choix qui ont été faits et une description minutieuse de l'interface que nous implémenterons.

Au chapitre quatre, nous détaillons les principes d'implémentation avec une description des fichiers, de l'architecture générale du programme et des différents modules.

Enfin, nous présentons, dans une cinquième partie, un manuel d'utilisateur et un exemple d'utilisation du traitement de texte.

I. ETUDE GENERALE

Dans ce premier chapitre, nous allons définir le problème de la création de documents à l'aide d'un traitement de texte.

Nous commencerons par préciser le concept de traitement de texte et certaines notions essentielles y afférant.

Ensuite, nous illustrerons le processus de création d'un document en spécifiant le système et le modèle de création d'un document.

Enfin, nous préciserons le système utilisé par la suite et nous présenterons aussi les spécificités d'un traitement de texte pour cadres supérieurs.

CHAPITRE I

1. QUESTIONS DE VOCABULAIRE

1.1. La terminologie

La terminologie concernant la manipulation des textes par ordinateur est très riche mais il n'existe pas, actuellement, de consensus bien établi. Essayons de faire le point sur le vocabulaire existant et tentons de définir un cadre de référence pour notre mémoire.

Suivant des possibilités fonctionnelles croissantes, on peut distinguer :

EDITEUR DE TEXTE

Ce que les anglo-saxons appellent "a text editor" est un des composants de base d'un système de traitement de texte, dont l'objet est la création et la maintenance d'un texte.

TRAITEMENT DE TEXTE

Ce que les anglo-saxons appellent "a text- processing" ou encore "a word processing" comprend un éditeur de texte et un formateur de texte c'est-à-dire un programme prenant en charge la présentation et la typographie du texte.

Souvent, les programmes de traitement de texte comportent en plus des programmes utilitaires pour analyser ou préparer un texte (ex : "spelling corrector" : programme de correction de fautes d'orthographe d'usage).

SYSTEME BUREAUTIQUE

Ce que les anglo-saxons appellent "an office automation system" combine typiquement un système de traitement de texte avec d'autres programmes utiles à un travailleur de bureau comme un gestionnaire de bases de données, un courrier électronique, un gestionnaire d'agenda, ...

CHAPITRE I

1.2. Notions de base

Nous allons introduire ici, quatre notions essentielles. Les deux premières sont fondamentales et seront développées abondamment dans les chapitres deux et trois ; les deux autres sont plutôt des éléments ou des vues concrètes du processus de création d'un texte.

Un MODELE est un cadre de référence abstrait du concepteur sur lequel le traitement de texte et le "monde" dans lequel il opère sont basés.

L'INTERFACE est l'intermédiaire entre l'ordinateur et ses périphériques - qui permettent la communication entre l'ordinateur et son environnement ; celui-ci nécessite du matériel, du logiciel ou, encore, les deux.

L'ECRAN est un quadrillage où chaque case permet de représenter un caractère. D'une façon plus imagée, l'écran peut être considéré comme une fenêtre placée sur une partie du texte. Le texte est un "parchemin" sur lequel l'écran donne une vue partielle.

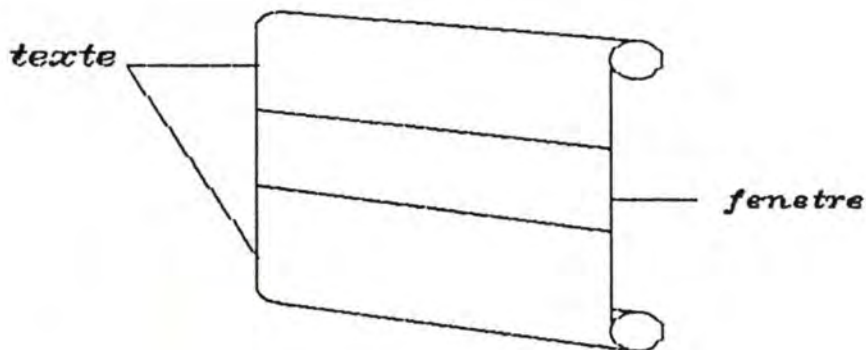


fig. 1.1 : Vue d'un document dans un système de traitement de
texte

Le curseur est une forme lumineuse sur l'écran vidéo qui indique l'endroit auquel le texte sera introduit. Celui-ci prend souvent la forme d'un petit carré clignotant.

CHAPITRE I

2. ETUDE DU PROCESSUS DE CREATION D'UN DOCUMENT

2.1. Constitution matérielle d'un système de traitement de texte

Comme nous l'avons représenté ci-dessous (fig. 1.2) , un système de traitement de texte comprend un ensemble de blocs fonctionnels spécialisés dans l'acquisition, la visualisation, la mémorisation et l'impression de texte. Ces divers blocs sont groupés autour d'une entité logique de traitement : l'unité centrale.

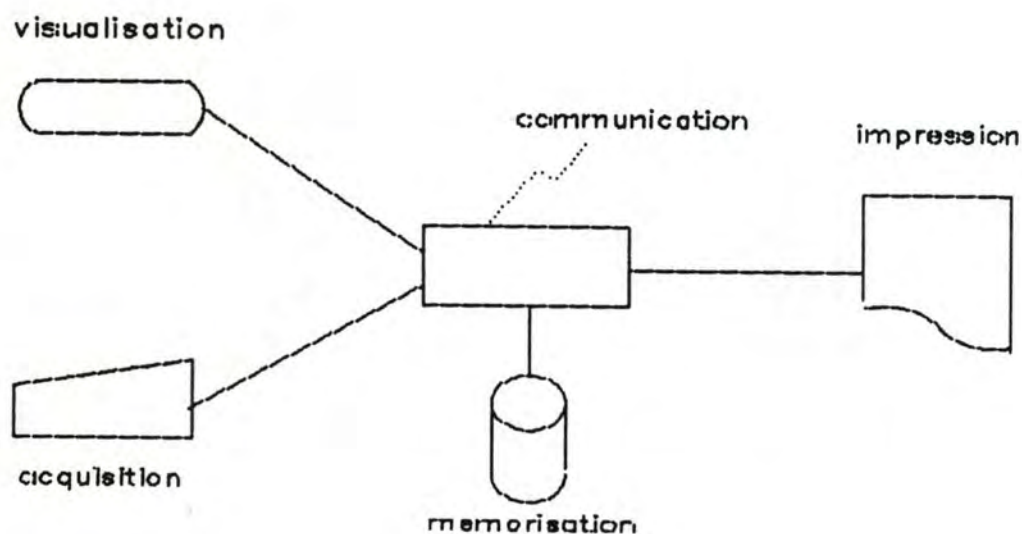


fig. 1.2 : Modèle général d'un système de traitement de texte

1. Acquisition d'informations

Parmi les dispositifs propres à réaliser l'acquisition d'information, le clavier est celui qui a été retenu.

CHAPITRE I

2. Visualisation d'informations

Cette fonction permet de fournir un support visuel d'information à l'utilisateur :

- en cours de saisie
- pendant la mise au point : après la saisie et avant l'impression sur papier. L'écran est le dispositif de visualisation d'informations qui a été retenu.

3. Stockage ou conservation d'informations

Il est réalisé au moyen d'un support externe permettant une mémorisation des textes. Le dispositif retenu a été le disque souple (tout autre support est aussi facilement envisageable).

4. Impression d'informations

C'est finalement le but de toutes les opérations de traitement de texte : une version sur papier du document. Le dispositif retenu a été l'imprimante.

5. Unité centrale de traitement

- unité spécifique de traitement
- mémoire

2.2. Modèle sous-jacent à la production d'un document

Le modèle devrait permettre de bien cerner les différentes étapes qui serviront de fil conducteur à l'analyse et même à l'implémentation du traitement de texte.

Si l'on observe le processus de production d'un document, les opérations élémentaires devraient s'enchaîner comme suit :

CHAPITRE I

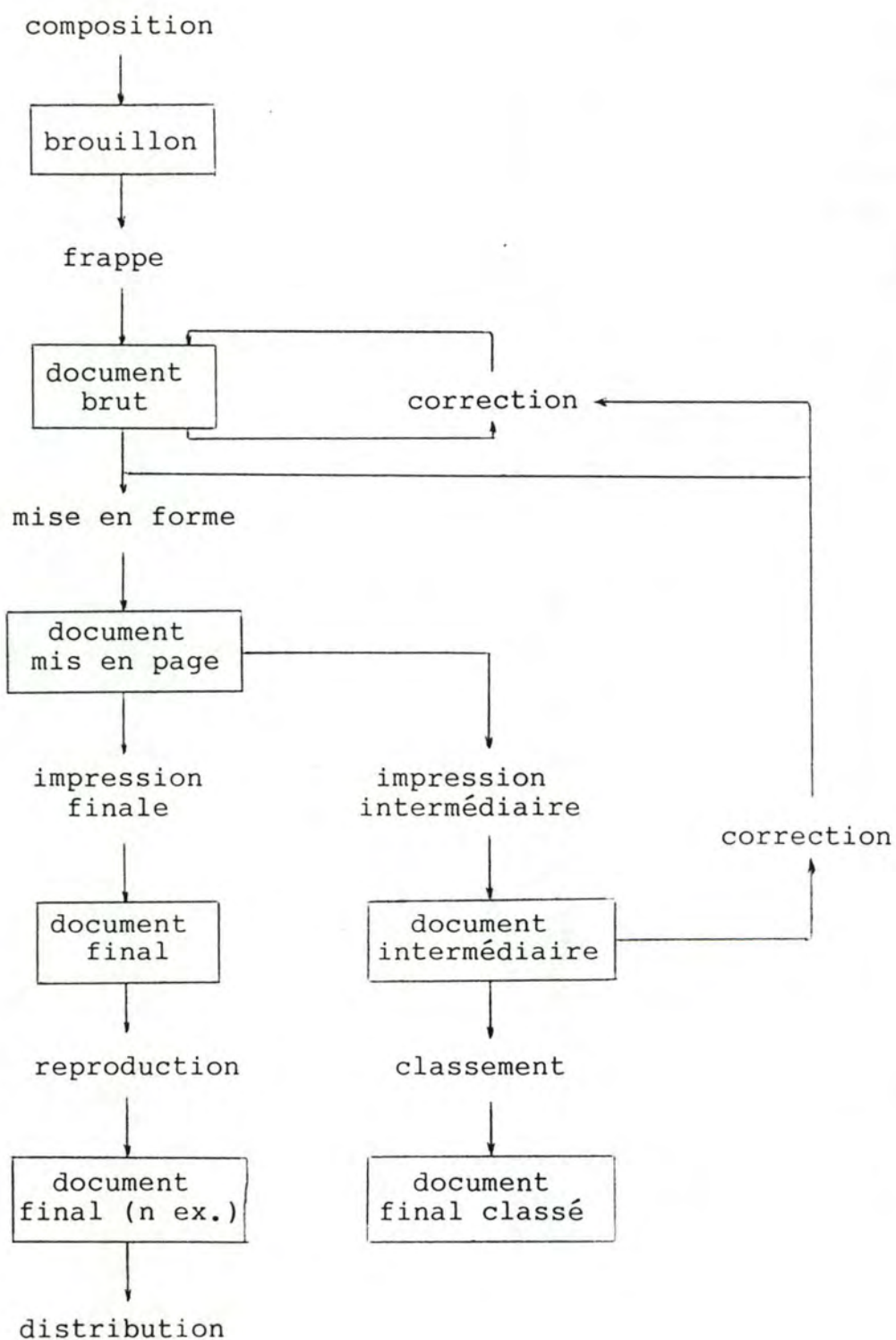


fig. 1.3 : Modèle général du flux d'opérations nécessaires à la production de documents

3. LE TRAITEMENT DE TEXTE POUR CADRE SUPERIEUR

3.1. Définition du système

Pour envisager les fonctions de base du traitement de texte et en cerner les possibilités, nous ferons toujours référence au système suivant (le système réel sur lequel sera implémenté le traitement de texte peut en être déduit facilement) :

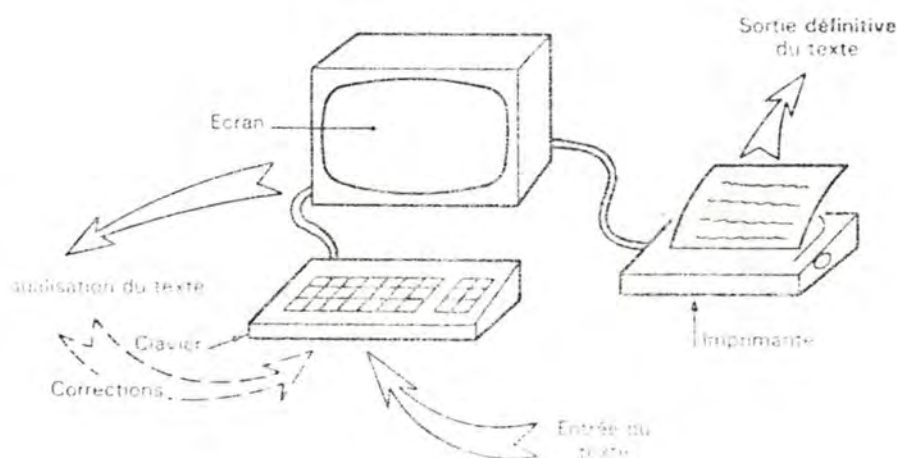


fig. 1.4 : Un système de traitement de texte

- clavier : pour introduire le texte
- imprimante : pour fournir un état imprimé
= machine à écrire traditionnelle
- écran : où apparaît le texte en cours d'élaboration
- mémoire : pour archiver et classer les textes

C'est l'écran et la mémoire qui sont spécifiques au traitement de texte. L'écran inclut la possibilité d'existence d'un brouillon privilégié qui permet des essais rapides et multiples avant une impression définitive. La mémoire permet de garder des copies non papiers des textes rédigés.

3.2. Spécificité du traitement de texte pour cadre supérieur

3.2.1. POURQUOI UN NOUVEAU TRAITEMENT DE TEXTE ?

Nous présentons, dans cette partie, les deux éléments qui justifient la réalisation d'un nouveau traitement de texte. Ils constituent, en fait, les lignes directrices de notre travail.

a) L'interface

On peut dire que l'on n'a pas encore épuisé le sujet. Beaucoup de choses existent mais il en reste encore à trouver.

Dans leur rapport (Meyrowitz, 2, p.402), N. Meyrowitz et A. Van Dam expliquent en ces termes l'état de la recherche en matière de conception des éditeurs de texte :

" Tandis que beaucoup de choses sont écrites à propos de " l'interface désirable homme/machine", la plupart étant souvent des opinions personnelles (non soutenues, non confirmées), très peu d'expériences déterminant l'interface optimal d'un éditeur ont été réalisées ".

Donc du côté de l'interface, il reste beaucoup à faire. Depuis Meyrowitz évidemment beaucoup d'efforts ont été dépensés au niveau de la recherche. C'est ainsi que l'on a vu apparaître des produits tels que "Star System" de Xerox et pour des applications plus restreintes, mais dans la même ligne d'idée, le "Macwrite" de Macintosh chez Apple. Du côté de l'interface, ces produits constituent "une révolution" en ce sens qu'ils introduisent un concept nouveau : le BUREAU ELECTRONIQUE (cfr. infra chap III 1.2).

b) Le modèle.

Les techniques de conception posent un autre problème (Meyrowitz, 2, p.402-403). Beaucoup d'éditeurs écrits aujourd'hui ont été conçus PAR des programmeurs POUR des programmeurs, et ont été commercialisés dans le grand public avec apparemment peu de soucis de leurs besoins. Beaucoup d'autres apparaissent comme conçus PAR des non-programmeurs POUR des non-programmeurs et présentent dès lors peu de caractéristiques telles qu'une "propre" fabrication de logiciel et des principes de conception bien définis tels qu'un ensemble consistant de commandes, une syntaxe consistante.

CHAPITRE I

Peu d'éditeurs ont été conçus après un examen rigoureux des choix raisonnables au niveau interface et fonctionnalité, et même peu sont supportés par un modèle conceptuel bien défini.

Il est donc temps, que les concepteurs d'éditeur et de traitement de texte décrivent d'abord le modèle conceptuel et les hypothèses retenues pour l'interface utilisateur avant toute implémentation.

3.2.2. TRAITEMENT POUR CADRE SUPERIEUR

Ces spécificités vont évidemment être un des buts de ce mémoire : il faudra sans cesse les avoir en arrière-pensée lors de la lecture du mémoire car bon nombre de décisions ont été prises en fonction de ces caractéristiques.

Bien entendu, le but n'est pas de tout faire c'est-à-dire, d'offrir un maximum de fonctions les plus puissantes possible, utilisables de façon la plus facile et la plus agréable possible : d'ailleurs, ces deux facteurs semblent antinomiques.

Nous retiendrons ici une idée-clé entendue récemment lors d'un symposium consacré au traitement de texte (Aiesec, 3) : un bon traitement de texte (BTT) est fonction de sa performance mais aussi de son acceptabilité :

$$\text{BTT} = f(\text{performance}, \text{acceptabilité})$$

On pense souvent au premier argument mais le second est souvent négligé.

Caractéristiques à retenir

Pour définir un cadre de référence au traitement de texte pour cadres supérieurs nous retiendrons les caractéristiques suivantes :

- 1) **Type de documents à produire.**
Ils sont toujours très courts (en moyenne 3 à 4 pages).
Pour des textes plus longs, le cadre supérieur fait appel à ses secrétaires.

CHAPITRE I

- 2) L'initiation au traitement de texte doit être minimale car un cadre supérieur n'a pas le temps (ni sans doute la volonté) de passer des heures d'initiation afin de pouvoir se servir du traitement de texte. Un cours de quelques minutes devrait-être suffisant pour rédiger un document : c'est le pari !
- 3) Pour rencontrer le but 2, il faut une manipulation aisée, naturelle du document. Si la manipulation est naturelle, il est alors possible de deviner ce que l'on peut faire dans une situation donnée même si à priori on ne le sait pas.
- 4) Dans l'optique que le temps c'est de l'argent (surtout le temps d'un cadre supérieur), des facilités comme suit seront les bienvenues :

- . on dispose de formulaires préremplis

Par exemple, un notaire peut travailler sur des contrats standards dans lesquels certaines clauses sont à compléter.

- . on dispose de formules/paragraphes pré-enregistrés et faciles à introduire dans un texte

Par exemple, le cadre supérieur dispose, pour la rédaction d'un contrat quelconque, de clauses standards qui peuvent être introduites rapidement.

- . le publi-postage (cfr "mailing" chez anglo-saxons) sera possible

Par exemple, pour inviter les membres d'un service à une réunion, le responsable peut envoyer une convocation en une seule opération.

On ne veut donc pas un traitement de texte qui fait tout mais un traitement qui fait bien (facilement-rapidement) ce qu'un cadre supérieur veut faire.

II. MODELE CONCEPTUEL D'UN DOCUMENT

Dans ce chapitre, nous allons d'abord définir un modèle des données : nous définirons un document, sa représentation ainsi que les différents types de documents que nous envisagerons.

Ensuite, nous spécifierons, sur base des besoins de l'utilisateur visé et du modèle de production d'un document, l'ensemble des fonctions souhaitées en décrivant leurs effets, sans nous soucier de la manière de les implémenter ; celles-ci seront présentées dans un premier classement établi selon les étapes de production d'un document.

CHAPITRE II

1. MODELE CONCEPTUEL DES DONNEES

1.1. Définition d'un document

Un document est une quantité d'informations écrites à l'aide du traitement de texte (lettre, formulaire, document quelconque) et destinées à être traitées.

Nous retiendrons trois concepts caractérisant un document :

- son CONTENU
- sa STRUCTURE LOGIQUE
- sa STRUCTURE PHYSIQUE

Le contenu est l'ensemble des signes (caractères) issus d'un alphabet (ensemble des touches d'un clavier) et destiné à exprimer une information.

Les structures sont des expressions différentes d'un MEME contenu. Nous distinguerons la structure

logique dont le but est d'accroître la facilité de compréhension du contenu informationnel

physique dont le but est de visualiser la structure logique sur une feuille ou un écran

1.2. Le modèle conceptuel

A l'utilisateur d'un système interactif de traitement de texte, on présente un MODELE CONCEPTUEL du système.

Le modèle conceptuel doit répondre aux questions suivantes :

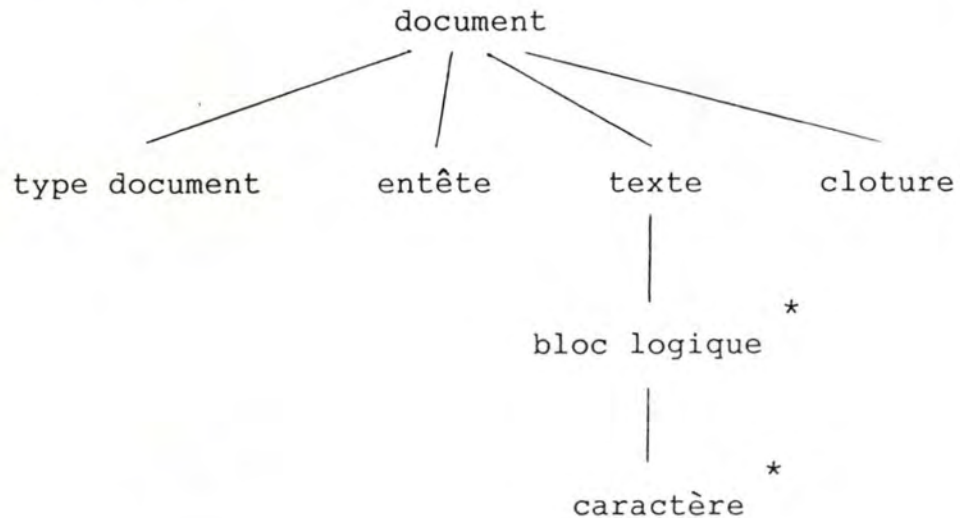
- quels sont les composants d'un document ?
- comment sont structurés les composants ?
- quelles sont les propriétés et les relations des composants ?

CHAPITRE II

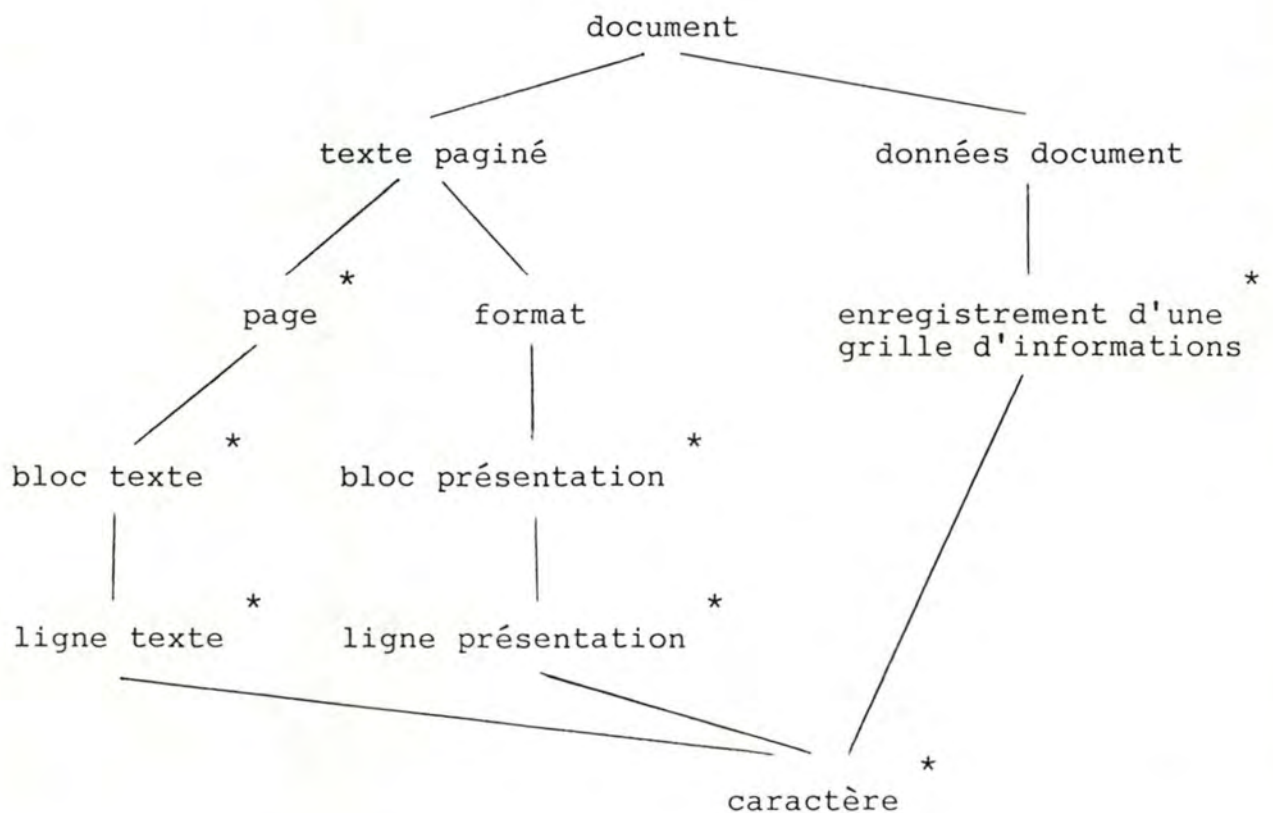
1.2.1. LES STRUCTURES D'UN DOCUMENT

Tout document a une double structure :

a) structure logique



b) structure physique



CHAPITRE II

1.2.2. DEFINITION DES ELEMENTS DE CES STRUCTURES

a) Composants de la structure logique

note : le signe "*" apparaissant dans les structures représente la cardinalité de l'élément structurel en question.

Par exemple, on lira qu'un document est composé de 0,1 ou plusieurs page(s).

1. TYPE DOCUMENT

Le composant "type document" est un indicateur du type du document.

Les types possibles pour un document sont au nombre de quatre :

- lettre (cfr. chap II 1.3)
- lettre standard (cfr. chap II 1.3)
- formulaire (cfr. chap II 1.3)
- quelconque (cfr. chap II 1.3)

2. ENTETE

L'entête d'un document comprend les attributs qui sont spécifiés ci-dessous :

- nom du document :
en vue de la manipulation, de l'impression ou de l'archivage, tout document aura un nom qui sera une suite de 10 caractères quelconques au maximum.

Ce nom sera identifiant et obligatoire.

- sujet :
le sujet d'un document peut être enregistré par un maximum de 4 mots-clés dont la longueur maximale est de 15 caractères quelconques mais il est facultatif.
- expéditeur :
le nom de l'auteur du document ou de celui qui l'envoie peut être représenté par une suite de 10 caractères quelconques au maximum mais il est facultatif.

CHAPITRE II

- destinataire :

représente le destinataire d'un document ; il est décrit par

- . nom : suite de 30 caractères quelconques au maximum
- . qualité : suite de 30 caractères quelconques au maximum
(exemple : administrateur délégué,
directeur général, ...)
- . adresse/numéro : suite de 30 caractères quelconques au
maximum
- . code postal/localité : suite de 30 caractères quelconques
au maximum

Toutes ces données sont facultatives et elles peuvent être pré-enregistrées dans un fichier d'adresses préalablement créé afin d'être automatiquement saisie (cfr. chap IV 1.4).

- date d'émission :

la date d'envoi du document (ou toute autre signification - ex: date d'écriture) est représentée par une suite de 10 caractères ; la structure de cette suite est bien précise :

caractère 1-2 jour
3 séparateur quelconque
4-5 mois
6 séparateur quelconque
7-10 année

où jour, mois et année doivent correspondre à une date du calendrier usuel.

La date d'émission a une valeur par défaut qui est la date du système.

3. CLOTURE

La clôture d'un document comprend les attributs suivants :

- date de réception :

la date de réception d'un document à la même structure que la date d'émission.

La date de réception a une valeur par défaut qui est la date du système.

- état :

représente l'état dans lequel se trouve un document.

Un document peut être . révisable
. revu
. envoyé
. classé.

CHAPITRE II

La sémantique même de ces états peut être définie par l'utilisateur ; on entend général et respectivement qu'un document peut être non signé, signé, que le destinataire peut en avoir connaissance, qu'il peut être archivé.

L'état d'un document a une valeur par défaut qui est "révisable" (état le plus primitif).

4. TEXTE

Le texte d'un document est constitué d'une suite de bloc(s) logique(s) (cfr. infra).

5. BLOC LOGIQUE

Le bloc logique est une structure de manipulation, une suite de caractères de longueur quelconque (au maximum le texte dans son entièreté) délimitée par deux caractères marqués (cfr. infra).

Cette notion regroupe, à elle seule, les notions habituelles de paragraphes, phrases, mots ; on ne considère qu'une seule entité logique qui est définie par deux caractères spéciaux de début et fin de blocs (cfr. marques).

6. CARACTERE

C'est l'unité logique de base ; c'est en fait le CONTENU sur lequel repose la structure logique.

Tout caractère peut être :

- . souligné ou non
- . une marque de bloc logique (en vue d'un effacement)
- . une marque de bloc texte (en vue de réduire l'interligne).

Par abus de langage, nous utiliserons fréquemment le terme "marque" pour désigner un caractère marqué.

b) Composants de la structure physique

1. DOCUMENT

(cfr. chap II 1.1)

CHAPITRE II

2. FORMAT

La définition de ce composant consiste en fait à spécifier un ensemble de valeurs que l'on appelle habituellement, les attributs de formatage ; c'est sur ces attributs que repose la structure physique du document.

- . **marge gauche :**
nombre entier représentant la position du premier caractère d'une ligne ; la valeur par défaut est 1.
- . **marge droite :**
nombre entier représentant la position la plus à droite possible d'une ligne à laquelle un caractère peut se trouver ; la valeur par défaut est 80.
- . **marge haut :**
nombre entier représentant le nombre de lignes de présentation laissées en haut d'une page avant la première ligne de texte ; la valeur par défaut est 3.
- . **marge bas :**
nombre entier représentant le nombre de lignes de présentation laissées en bas d'une page après la dernière ligne de texte ; la valeur par défaut est 3.
- . **longueur page :**
nombre entier représentant le nombre de lignes de présentation et de texte qui seront imprimées sur une page (cfr. infra) ; la valeur par défaut est 66.
- . **interligne :**
attributs à valeur booléenne indiquant si le texte d'un document doit être imprimé en interligne réduit ou non ; la valeur par défaut est l'interligne simple.

un texte peut être imprimé en interligne simple (1/6 de pouce) ou en interligne réduit (7/72 de pouce).

3. TEXTE PAGINE

C'est un ensemble de pages (cfr. infra).

CHAPITRE II

4. PAGE

Une page est l'unité physique représentant l'arrangement des signes (caractères) imprimables sur une feuille de papier ou sur un écran ; cet arrangement est spécifié par les valeurs des attributs de formatage qui sont définis par le composant "FORMAT" (cfr. supra).

5. BLOC TEXTE

C'est une suite de ligne(s) de texte (cfr.infra).
délimitée par deux caractères marqués.

Un bloc texte peut être en interligne réduit ou non (cfr. supra).

6. BLOC PRESENTATION

C'est une suite de ligne(s) de présentation (cfr.infra).

7. LIGNE TEXTE

C'est une suite de caractères qui est comprise entre la marge gauche et le caractère délimiteur de fin de ligne.
Elle représente les informations contenues dans le texte d'un document.

8. LIGNE PRESENTATION

C'est une ligne texte qui a pour but de représenter le contenu informationnel du format (marge haut et marge bas) et de l'entête (destinataire).

C'est en fait les lignes blanches laissées en bas et en haut d'une feuille de papier et les lignes contenant l'adresse d'une lettre.

(rem : quand on parlera de ligne, il s'agira indifféremment d'une ligne texte ou d'une ligne présentation)

9. CARACTERE

C'est l'unité physique de base ; c'est en fait le CONTENU sur lequel repose la structure physique.

CHAPITRE II

10. DONNEE DOCUMENT

C'est un ensemble d'enregistrement(s) (cfr.infra).

11. ENREGISTREMENT D'UNE GRILLE D'INFORMATION

C'est l'unité d'information reflétant le contenu informationnel de format, de entête, de clôture et de type document.

C'est, en fait, toutes les données annexes au texte d'un document telles que son auteur, son destinataire, ...

1.3. Les différents types de document

Nous considérerons différents types de document auxquels correspond une structure ou un sens particulier :

1.3.1. Lettre

structure :

- entête : nom/sujet/expéditeur/destinataire/date d'émission
- texte
- clôture : date réception/état

sens :

représente une lettre au sens habituel du terme.

1.3.2. Lettre Standard

structure :

même structure qu'une lettre

sens :

représente une lettre obtenue à partir d'un formulaire (cfr. infra).
C'est, en fait, un formulaire rempli.

1.3.3. Formulaire

structure :

- entête : nom/date d'émission

sens :

représente un formulaire au sens habituel du terme c-à-d un texte qui comprend des trous (cfr. lexique).

1.3.4. Quelconque

structure :

- entête : nom/date d'émission

sens :

représente un document qui n'est pas un des trois types précédemment définis.

CHAPITRE II

2. LES FONCTIONS DU TRAITEMENT DE TEXTE

Suite à l'analyse des besoins des cadres supérieurs tels qu'ils ressortent d'une enquête menée par la Standard Oil Company (Indiana) sur "Office Technology Systems Conceptual Fonctionnal Requirements from a User's Perspective" (S.O.C., 4) et à partir du modèle sous-jacent à la production de documents, nous avons retenu les fonctions suivantes que devra réaliser un traitement de texte adéquat.

2.1. Fonctions destinées à faciliter l'opération de première frappe

Nous allons décrire ci-dessous, l'ensemble des fonctions nécessaire et suffisant pour permettre à l'utilisateur de saisir un texte :

1. POSE DES MARGES ET TABULATIONS

Comme sur une machine à écrire, la première chose que l'on fait avant d'introduire un texte quelconque, est de régler les marges et de définir des tabulations si celles qui sont présentes ne sont pas satisfaisantes. On définit donc les limites extrêmes gauche et droite d'une ligne (donc, on définit la notion de ligne), et on installe des repères utiles sur cette ligne pour la saisie du texte. Elles peuvent être modifiées n'importe quand.

2. ENREGISTREMENT DE CARACTERES

Afin de pouvoir saisir un texte, il est nécessaire de pouvoir enregistrer des caractères. L'enregistrement d'un caractère ne nécessite rien d'autre que de le frapper sur le clavier : initialement, il sera enregistré au début du texte, puis, de façon répétée, à la suite du dernier caractère enregistré.

L'enregistrement d'un caractère en fin d'écran provoque un décalage du document sur l'écran afin de visualiser le caractère que l'on enregistre.

3. CORRECTION DES FAUTES DE FRAPPE

Suppression d'un caractère

Suite à l'introduction d'un caractère erroné, l'utilisateur doit être capable d'apporter une correction directement afin d'éliminer ce mauvais enregistrement. De même, l'opération peut être exécutée de façon répétée afin de supprimer des mots entiers voire une ligne entière.

Insertion d'un caractère

Suite à l'omission de caractères, l'utilisateur peut être amené à apporter une correction. Ce genre de correction est fonctionnellement similaire à l'enregistrement de caractère mis à part le fait que son domaine d'applicabilité est différent : alors que l'enregistrement s'effectue toujours en fin de texte, l'insertion sera l'addition d'un ou de plusieurs caractère(s) à l'intérieur d'un texte précédemment enregistré ou inséré.

De même, l'opération peut être exécutée de façon répétée afin d'insérer des mots entiers, des lignes entières, voire plusieurs lignes.

Remplacement d'un caractère par un autre

Suite à l'introduction d'un caractère erroné, l'utilisateur doit être capable d'apporter une correction sans devoir explicitement supprimer un caractère et insérer le caractère correct. Il y a simplement une substitution entre les deux caractères.

4. SOULIGNEMENT D'UN CARACTERE

Le soulignement est entendu sous deux formes différentes. Il est possible de souligner des caractères figurant déjà dans le texte (il s'agit du sens commun du soulignement comme on l'entend sur une machine à écrire), mais l'utilisateur peut introduire des caractères soulignés dans le texte.

Nous permettrons également à l'utilisateur de supprimer un soulignement sans l'obliger à effacer le caractère souligné.

CHAPITRE II

5. "RETOUR CHARIOT" AUTOMATIQUE EN FIN DE LIGNE SANS CESURE DE MOTS

C'est ce que les anglo-saxons appellent le "word wrapping". Il s'agit du déplacement automatique d'un mot à la ligne suivante, si le mot ne peut être contenu entièrement dans la ligne courante. De cette façon, un mot n'est jamais coupé en deux à moins que l'utilisateur ne le fasse explicitement. Si le texte à introduire excède la longueur de la ligne, le système de traitement de texte le place sur la ligne suivante : il plie donc la ligne courante sur la ligne suivante.

Le gros avantage est la possibilité de frapper d'un document sans tenir compte de la structure des lignes. En fait on frappe les mots les uns derrière les autres en laissant à la machine le soin de les organiser en lignes. Ce type de frappe est habituellement désigné par le terme "saisie de texte au kilomètre".

6. MEMORISATION DU TEXTE

Il s'agit, lorsque l'on a terminé la saisie de texte, d'enregistrer sur une unité de mémorisation à long terme (disque souple ou disque dur), le travail effectué.

Les trois fonctions qui suivent s'insèrent dans le cadre de l'édition rapide et facile de texte qui est un critère de base de ce traitement de texte.

7. ENREGISTREMENT DE TEXTE STANDARD

Cette fonction permet d'introduire rapidement du texte sans devoir le retaper explicitement. Les textes qui sont introduits de cette façon sont en général n'importe quelle suite de caractères mais, on retient habituellement deux cas bien particulier d'utilisation :

- pour permettre d'insérer rapidement des éléments répétitifs intervenant à plusieurs reprises dans le document (possibilité d'abréviations)
- permettre de disposer d'une bibliothèque de paragraphes formules de politesse diverses suivant l'humeur du moment ou le destinataire de la lettre

CHAPITRE II

8. REMPLISSAGE DE FORMULAIRE

C'est la possibilité d'utilisation d'un squelette de document pré-enregistré (formulaire) et remplissage des trous.

9. PUBLI-POSTAGE

Il s'agit de ce qu'on entend couramment sous le terme de "mailing" dans la littérature spécialisée : à partir d'un document et d'une liste d'adresses pré-enregistrée, il est possible de faire une fusion des deux afin de produire autant de documents personnalisés qu'il n'y a d'adresses.

2.2. Fonctions destinées à faciliter la correction d'un document

Par exemple, lors de la relecture d'un document précédemment mémorisé, on est bien souvent amené à modifier son contenu en ajoutant ou en supprimant des caractères, des lignes ou encore n'importe quelle suite de caractères. Pour manipuler ce caractère ou cette suite, l'utilisateur dispose outre de certaines fonctions déjà spécifiées (cfr. 1, 2, 3, 4, 5, 7), de fonctions spécifiques à cet usage telles :

1. OUVERTURE DE FENETRE DANS LE TEXTE

Il s'agit d'écarter le texte verticalement, afin d'avoir sous la ligne désirée une ligne blanche. L'écartement se fait ligne par ligne et l'utilisateur peut écarter le texte autant qu'il le veut afin soit de l'aérer, soit d'insérer du nouveau texte.

2. RECHERCHE D'UNE OCCURRENCE DE CARACTERES

Il s'agit de rechercher dans le texte une occurrence de caractères donnée afin, par exemple, d'opérer, à cet endroit, une insertion.

En fait, il s'agit d'un déplacement rapide du curseur jusqu'à un endroit spécifié ; si celui-ci n'existe pas, la commande a pour effet de signaler l'absence de cette occurrence de caractères, sans autre effet.

CHAPITRE II

3. EFFACEMENT JUSQU'EN FIN DE LIGNE

Cette fonction a pour effet d'effacer la ligne désirée, à partir de l'endroit souhaité, jusqu'en fin de ligne. Si l'endroit souhaité est la fin de ligne, la commande est sans effet.

4. EFFACEMENT D'UNE LIGNE

Cette fonction efface une ligne désirée : cela signifie que le contenu de la ligne va être supprimé mais aussi que la ligne elle-même est supprimée (ré-ajustement du texte : cfr. lexicque).

5. POSE D'UNE MARQUE DE BLOC

Les marques sont des caractères du texte que l'utilisateur définit comme caractère marqué. Elles permettent de définir un bloc logique qui est soit une fin de ligne de texte (une seule marque), soit n'importe quelle suite de caractères pouvant même s'étendre sur plusieurs lignes (deux marques).

6. EFFACEMENT D'UN BLOC.

En plus des possibilités d'effacement déjà spécifiées, on est parfois appelé à détruire d'importantes quantités de texte. Il s'agit alors effacer un bloc logique préalablement défini.

L'exécution de cette commande demande au préalable d'avoir marqué le bloc à effacer.

2.3. Fonctions destinées à faciliter l'opération de mise en forme

Une fois qu'un document est écrit, relu et corrigé, il sera probablement imprimé. Cependant, auparavant, il doit être préparé pour l'impression. Par exemple, les marges, la longueur de la page doivent être données au système : c'est le formatage. Il s'agit, en fait d'arranger le texte selon des instructions spécifiques.

Pour analogie, en dactylographie, des exemples d'instructions de formatage sont le réglage de la marge, les taquets de tabulation et la longueur de la page.

CHAPITRE II

Dans le traitement de texte, le formatage inclut également la justification.

1. DEFINITION ET MODIFICATION DE L'INTERLIGNE

L'interligne peut être défini de deux façons :

- choix d'un interligne unique pour l'entièreté du document ; l'interligne peut être simple ou réduit
- choix d'un interligne pour un certain nombre de lignes que l'utilisateur spécifie lui-même (définition d'un bloc texte).

2. JUSTIFICATION A DROITE

Le texte peut être, dans son entièreté, justifié à droite c'est-à-dire que sur chaque ligne, le dernier caractère se trouve sur la marge droite et n'est pas un blanc.

Cependant, les lignes de fin de paragraphe ne sont pas justifiées, ni celles qui ne sont pas suffisamment longues (il est préférable d'avoir une ligne non justifiée et facilement lisible que d'avoir une ligne justifiée mais devenue illisible à cause d'un étirement horizontal abusif).

Après une justification, le texte est aligné verticalement à gauche (celui-ci se fait automatiquement lors de la saisie initiale du texte) et à droite pour former un rectangle parfait.

3. CENTRAGE D'UNE LIGNE

Une ligne de texte peut-être centrée par rapport aux deux marges définies.

4. DEFINITION DES PARAMETRES DE MISE EN PAGE

- marge gauche : cfr. chap II 1.2.2.b
- marge droite : cfr. chap II 1.2.2.b
- marge haut : nombre de ligne(s) blanche(s) désirée(s) en début d'une page avant la première ligne du document sur cette page.
- marge bas : nombre de ligne(s) blanche(s) désirée(s) en fin d'une page après la dernière ligne du document sur cette page.

CHAPITRE II

5. PAGINATION AUTOMATIQUE

Il s'agit de découper le document en un ensemble de pages dont le format a été défini préalablement.

L'utilisateur pourra apprécier sur l'écran la mise en page de son document avant par exemple d'en demander le produit fini, à savoir, une copie papier.

2.4. Fonctions destinées à faciliter la manipulation de documents

1. AFFICHAGE DU REPERTOIRE

Il s'agit de l'affichage du répertoire (liste de) des documents enregistrés sur une unité de mémorisation à long terme (disque souple ou disque dur).

Le répertoire reprend le nom des documents enregistrés, la date de leur dernier enregistrement et l'auteur (expéditeur) de ce document.

2. RECHERCHE DE DOCUMENT SUR MOT-CLE

La recherche, sur base d'une référence - en l'occurrence un mot-clé -, d'un document enregistré permet d'obtenir, par exemple, la liste de tous les documents qui traitent de tel sujet particulier.

3. EFFACEMENT D'UN DOCUMENT

Il s'agit de supprimer un document enregistré dont on n'a plus besoin ; on le fait alors disparaître du support externe de mémorisation.

4. RENOMMER UN DOCUMENT

Il s'agit de modifier le nom d'un document.

2.5. Fonction destinée à faciliter l'impression d'un document

Il n'y a en fait ici besoin que d'une simple fonction pour commander une impression à l'imprimante car toutes les options possibles de mise en forme sont définies par des fonctions de mise en page et peuvent être ainsi visualisées sur l'écran avant l'impression physique du document.

2.6. Fonctions destinées à se déplacer dans le texte

Les déplacements du curseur sont, en fait, des déplacements à l'intérieur de tout le document quand bien même celui-ci ne tiendrait pas entièrement sur l'écran. Il faut, en effet, considérer que l'écran du système n'est qu'une fenêtre qui nous laisse voir une partie du texte global seulement. En déplaçant la fenêtre dans le texte et en déplaçant le curseur dans l'écran, toutes les parties du texte peuvent être atteintes.

2.6.1. FONCTIONS DE DEPLACEMENT DU CURSEUR DANS L'ECRAN

1. GAUCHE

Cette fonction permet, normalement, de se déplacer latéralement sur une ligne de l'écran vers la gauche : elle permet de reculer d'un caractère s'il existe.

En début de document (c'ad quand le curseur est positionné sur la marge gauche de la première ligne), la commande est sans effet.

2. DROITE

Cette fonction permet, normalement, de se déplacer latéralement sur une ligne de l'écran vers la droite : elle permet d'avancer d'un caractère s'il existe.

En fin de document (c'ad quand le curseur est positionné sur la marge droite de la dernière ligne ou lorsque le curseur se trouve juste après le dernier caractère introduit sur cette ligne, la commande est sans effet.

CHAPITRE II

3. HAUT

Cette fonction permet, normalement de se déplacer verticalement dans le texte du document vers le haut : elle permet d'aller à la ligne précédente si elle existe.

Lorsque le curseur se trouve sur la première ligne du document, la commande est sans effet. Si par contre, il est sur la première ligne de l'écran et que cette ligne n'est pas la première du document, un nouvel écran est affiché de telle manière que la ligne précédente soit affichée comme première ligne de l'écran.

4. BAS

Cette fonction permet, normalement de se déplacer verticalement dans le texte du document vers le bas : elle permet d'aller à la ligne suivante si elle existe.

Lorsque le curseur se trouve sur la dernière ligne du document, la commande est sans effet. Si par contre, il est sur la dernière ligne de l'écran et que cette ligne n'est pas la dernière du document, un nouvel écran est affiché de telle manière que la ligne suivante soit affichée comme dernière ligne de l'écran.

5. ALLER EN DEBUT D'ECRAN

Dans tous les cas, la commande déplace le curseur sur le premier caractère de la première ligne de l'écran courant.

6. ALLER EN FIN D'ECRAN

Normalement, le curseur est déplacé juste derrière le dernier caractère ou sur la marge droite de la dernière ligne de l'écran courant. Si l'écran contient la dernière ligne du document, la commande déplace le curseur sur le dernier caractère ou sur la marge droite de la dernière ligne du document.

CHAPITRE II

2.6.2. FONCTIONS DE DEPLACEMENT DE L'ECRAN DANS LE TEXTE

1. AFFICHER L'ECRAN SUIVANT

Cette fonction recherche la dernière ligne affichée à l'écran et la recopie comme première ligne d'un nouvel écran ; celui-ci est complété par les lignes suivantes du document et le curseur est placé dans le coin supérieur droit du nouvel écran.

Si l'écran courant est le dernier écran (càd la dernière ligne du document est affichée à l'écran, la commande est sans effet.

2. AFFICHER L'ECRAN PRECEDENT

Normalement, cette fonction recherche la première ligne affichée à l'écran et la recopie comme dernière ligne d'un nouvel écran ; celle-ci est précédée par les lignes précédentes du document. Si les lignes qui précèdent ne suffisent pas pour remplir un écran, la première ligne du document est alors placée comme première ligne de l'écran qui est alors complété par les lignes suivantes du document. Le curseur est placé dans le coin supérieur droit du nouvel écran.

Si l'écran courant est le premier écran (càd la première ligne du document est affichée à l'écran), la commande est sans effet.

III. CONCEPTION DE L'INTERFACE

Nous introduirons le chapitre par une critique sommaire de trois interfaces choisis de façon à représenter les différents types d'interface existants.

Ensuite, nous expliquerons les choix qui ont été faits pour notre interface et le décrirons de façon précise avec des exemples en guise d'illustration.

CHAPITRE III

1. ANALYSE DE DIFFERENTS TYPES D'INTERFACES

1.1. Wordstar (classique le plus populaire) (Virgatchick, 5) (Manuel, 6)

1.1.1. LES PERIPHERIQUES D'ENTREE

Sur le clavier, Wordstar possède plusieurs dizaines de commandes (des centaines !). Face à cette abondance, une société anglaise (WMI Products) a même jugé utile d'ajouter à l'ordinateur un périphérique d'un genre spécial (KEYSTAR) pour faciliter l'apprentissage et l'utilisation du traitement de texte ; c'est un boîtier composé de 66 touches (en 4 couleurs) connectable à l'ordinateur : il se charge des séquences de commandes nécessaires à l'utilisation du traitement de texte. Chaque touche est dédiée à une fonction clairement indiquée en toutes lettres.

1.1.2. LANGAGE D'INTERACTION

Toutes les commandes sont reprises dans des MENUS DE COMMANDES. Wordstar affiche de nombreux menus qui, en principe, doivent aider l'utilisateur dans son apprentissage. Ces menus -dont l'idée est excellente- sont malheureusement fort complexes, mal structurés, et ne servent en réalité pleinement qu'au moment où l'utilisateur a déjà une excellente connaissance du logiciel.

1.1.3. L'ECRAN

Pour les débutants ou les personnes utilisant rarement Wordstar, l'écran est très encombré car les menus de commandes, toujours présents, occupent environ la moitié de l'écran. C'est seulement au cours d'un apprentissage progressif que les menus pourront être réduits, voire disparaître.

1.1.4. CONCLUSIONS

Wordstar, le traitement de texte certainement le plus populaire actuellement, n'est manifestement pas un outil pour cadre supérieur ; il n'a d'ailleurs pas cette prétention.

Très puissant pour qui sait l'utiliser et l'utilise fréquemment, Wordstar est assez difficile à apprendre - et le manuel d'utilisation (trop complet et redondant) ne facilite guère les premières découvertes.

CHAPITRE III

1.2. Macwrite (nouvelle tendance) (Van Loo, 7)

1.2.1. LES PERIPHERIQUES D'ENTREE

Le clavier est réduit à sa plus simple expression ; il ne ressemble en rien à un clavier conçu pour le traitement de texte avec de nombreuses touches fonctions ; c'est à peu de choses près un clavier de machine à écrire.

Par contre, un des atouts est l'utilisation intensive de la souris : petite boîte que l'on déplace manuellement afin de déplacer un pointeur sur l'écran qui sélectionnera les objets et actions.

1.2.2. L'ECRAN

Sa présentation est en fait l'originalité de Macwrite (et de tout les logiciels orienté objets). Pour ne pas dépayser l'utilisateur, l'écran ressemble à un bureau sur lequel on voit des objets familiers tels que des dossiers, des presses-papiers, une poubelle : c'est le concept de BUREAU ELECTRONIQUE.

L'inconvénient majeur est du au Machintosh lui-même étant donné la taille relativement réduite de son écran.

1.2.3. LANGAGE D'INTERACTION

Toutes les commandes sont exécutées par SELECTION D'OBJETS. Par exemple, on prend un dossier (on le pointe avec la souris), on en extrait un document que l'on ouvre (on sélectionne l'option ouvrir par pointage dans un menu que l'on fait apparaître de la même façon) afin d'afficher son contenu sur l'écran.

Il est conçu de telle sorte qu'il travaille comme le ferait un utilisateur.

1.1.4. CONCLUSIONS

Le concept de base du Macintosh (matériel sur lequel fonctionne Macwrite) est que un ordinateur est d'autant plus pratique qu'il est simple à utiliser. Le mode de fonctionnement du Macintosh a pour but de réduire au strict minimum la nécessité des modes d'emploi. Grâce aux fenêtres, icônes et autres souris, le traitement de texte Macwrite veut aussi respecter ce principe.

Il permet de visualiser les documents exactement tels qu'ils seront imprimés.

CHAPITRE III

1.3. Professional editor (le plus simple) (Manuel, 8)

1.3.1. LE CLAVIER

Il est très classique avec des touches fonctions.

1.3.2. L'ECRAN

Classique lui aussi mais l'utilisateur ne dispose pas de tout l'écran. Il y a les 9 premières colonnes qui sont toujours réservées comme zone d'entrée de commandes : il y a un gaspillage.

1.3.3. LANGAGE D'INTERACTION

Il se présente principalement sous deux formes. Des MENUS qui sont surtout utilisés pour la sélection de données et la PROGRAMMATION DES TOUCHES FONCTIONS utilisées pour spécifier les commandes. Il y a encore des commandes d'édition proprement dites (dans une ligne réservée) et des commandes lignes (dans les colonnes 1-9 de chaque ligne).

1.3.4. CONCLUSIONS

Pedit est facile à apprendre et à utiliser mais est fortement limité au niveau performance : pas de publi-postage, pas de mémorisation de paragraphes standards, pas de gestion des textes édités, ...

CHAPITRE III

2. DESCRIPTION DE L'INTERFACE

2.1. But de l'interface

A l'utilisateur d'un traitement de texte, on présente, grâce à un interface, la collection des outils et techniques grâce auxquels il peut communiquer avec le programme de traitement de texte (principalement les fonctions définies au chapitre précédent).

2.2. Description

L'interface utilisateur contient des éléments logiciels (le langage d'interaction du système) et des éléments matériels (les périphériques d'entrée et de sortie).

2.2.1. LES ELEMENTS LOGICIELS

Le langage interactif choisi est essentiellement constitué de :

a) Programmation des touches fonctions

C'est le principal langage d'interaction utilisé dans la phase de traitement du texte.

- *) Il y a en tout 29 fonctions programmées sur les 10 touches (F1 - F10) du clavier (cfr. fig. 3.1).

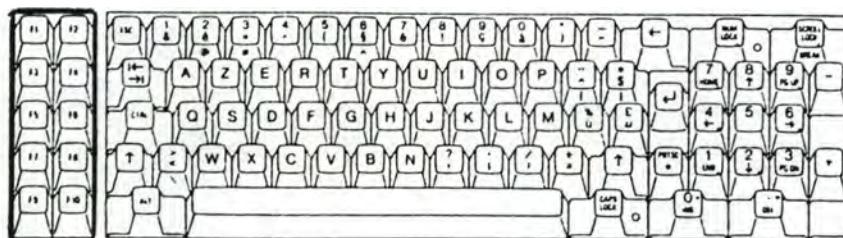


fig. 3.1 : Les touches fonctions sur le clavier

CHAPITRE III

Elles sont réparties en 5 jeux (0-4) suivant leur fonctionnalité afin de pouvoir être implémentées sur les 10 touches du clavier et rester cohérent ; on a, de cette façon, des touches claviers numérotées de 0 - 49.

Bien que cela ne sera vraisemblablement pas perçu par l'utilisateur (il n'a d'ailleurs pas à s'en soucier), fonctionnellement les touches sont réparties comme suit :

note :

le premier chiffre du nombre représente le jeu fonctionnel auquel la fonction appartient (une même fonction peut appartenir à des jeux différents mais si c'est le cas, c'est toujours pour une même touche clavier)

00 ANNUL	10 ANNUL	20 ANNUL	30 ANNUL	40 ANNUL
01 CREAT	11 +ADRES	21 MARQUE	31 MARQUE	41 NORMALE
02 MODIF	12 -ADRES	22 EFF-REG	32 OPTIONS	42 PUBLI-POST
03 MIS PAG		23 CHERCHE	33 MIS PAG	
04 IMPRES	14 +LIGNE	24 +LIGNE	34 +LIGNE	44 ADRESSE
05 MANIP	15 SOULIG	25 SOULIG	35 -LIGNE	45 REPERT
	16 SAUT		36 CENTRER	46 EFFACE
	17 DICT		37 JUSTIFIER	47 RENOMME
	18 TAB			48 REC DOC
	19 ENREG	29 ENREG	39 ENREG	

Les commandes ainsi implémentées ont l'objectif suivant :

Annul (00, 10, 20, 30, 40) : annule l'opération en cours
c'est à dire l'action courante qui est mentionnée dans la
ligne des messages (cfr. chap III 2.2.1.b).

Creat (01) : création d'un document.

Modif (02) : modification d'un document.

Mis pag (03, 33) : mise en page d'un document.

Impres (04) : impression d'un document.

Manip (05) : manipulation d'un document.

+Adres (11) : permet d'ajouter une adresse à une liste
existante ou non.

CHAPITRE III

- Adres (12) : permet d'éliminer une adresse d'une liste.
- +ligne (14, 24, 34) : créer une fenêtre d'une ligne blanche dans le texte en dessous de la position du curseur.
- Soulig (15, 25) : positionne le mode "soulignement" on/off.
- Saut (16) : permet le déplacement du curseur de trou en trou dans un formulaire (cfr. lexique).
- Dict (17) : permet d'utiliser le dictionnaire pour l'introduction rapide de texte.
- Tab (18) : permet de modifier les tabulations et les marges.
- Enreg (19, 2, 3) : permet d'enregistrer un document sur le support externe.
- Marque (21, 31) : indique qu'un caractère du texte est une marque d'un bloc logique ou d'un bloc texte.
- Eff - reg (22) : permet d'effacer une région c'est-à-dire une suite de caractères comprise entre deux caractères marqués.
- Cherche (23) : recherche dans le texte une occurrence d'une chaîne de caractères donnée à partir de la position courante.
- Options (32) : permet de faire apparaître le formulaire avec les options de formatage pour éventuellement les modifier.
- LIGNE (35) : permet de supprimer une ligne texte.
- Centrer (36) : permet de centrer une ligne texte.
- Justifier (37) : permet de justifier toutes les pages d'un document.
- Normale (41) : permet de commander une impression d'un document.
- Publi-post (42) : permet de commander une impression de plusieurs documents (cfr. def. publi- postage).
- Adresse (44) : permet la gestion des listes d'adresses.
- Repert (45) : permet de faire défiler sur l'écran le répertoire.
- Efface (46) : permet de supprimer un document du support externe.

CHAPITRE III

Renname (47) : permet de changer le nom d'un document.

Rec doc (48) : permet de rechercher la liste des documents qui possèdent un mot clé donné.

*) en plus des 10 touches claviers (F1-F10), il y a aussi :

- 8 touches fonctions programmées (clavier numérique en mode non numérique)

flèche à gauche	: reculer d'un caractère
flèche à droite	: avancer d'un caractère
flèche en haut	: monter d'une ligne sur l'écran
flèche en bas	: descendre d'une ligne sur l'écran
home	: aller au début d'un écran
END	: aller à la fin d'un écran
PG UP	: aller à l'écran précédent
PG DW	: aller à l'écran suivant

- 3 fonctions particulières

BS (backspace)	: effacer le caractère précédent la position courante
Del	: effacer le caractère à la position courante
Ins	: positionner les modes insertion/remplacement

- 1 touche de tabulation

(cfr. fig. 3.2)

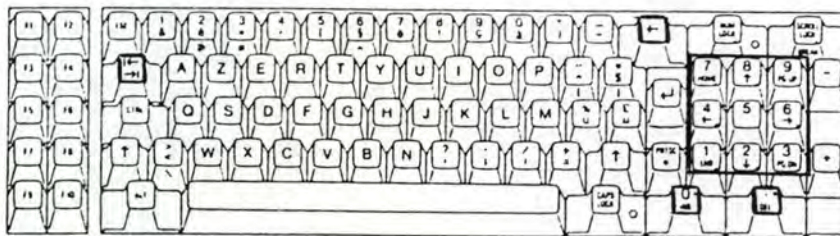


fig. 3.2 : Les touches spéciales du clavier

b) Fiches à remplir

C'est le principal langage d'interaction utilisé dans la phase de saisie des données.

En général, en plus du nom du document sur lequel on veut travailler, il y a d'autres renseignements qui sont demandés suivant le type d'action que l'on veut entreprendre.

Les trois fiches principales sont :

1) Saisie d'une lettre (fiche complète telle qu'elle apparaît à l'écran)

Nom de la lettre :	
Sujet :	
mot clé 1 :
mot clé 2 :
mot clé 3 :
mot clé 4 :
Expéditeur :	
Date d'émission :	
Destinataire :	
Nom :
Qual :
Adr, n° :
Cp, loc :

CHAPITRE III

2) Correction d'un document (fiche complète)

2a)

Nom du document :

Etat : révisable
 revu
 envoyé
 classé

Date de réception :

2b)

Expéditeur :

Destinataire :

Nom :

Qual :

Adr, n° :

Cp, loc :

Sujet :

mot clé 1 :

mot clé 2 :

mot clé 3 :

mot clé 4 :

3) Saisie des options de mise en page

Marge supérieure	(0...10)	: 3
Marge inférieure	(0...10)	: 3
Marge gauche	(1...25)	: 1
Marge droite	(55...80)	: 80
Longueur page	(10...66)	: 66
Interligne-réduit	(0...11)	: 11

En général, les autres fiches ne sont que des sous-ensembles d'une de ces trois-ci.

CHAPITRE III

Pour les données qui ont des valeurs déjà déterminées ou par défaut, les fiches à remplir sont optionnelles (ex : 3 et 2b).

De même, elles peuvent être minimisées par le choix du type du document lorsque celui-ci importe peu (choisir type de document "quelconque").

c) Ligne de présentation

Langage interactif utilisé pour modifier les tabulations et les marges gauches et droites d'un texte.

C'est une règle graduée (cfr. fig. 3.3) vue comme un message pour l'utilisateur lui signalant l'emplacement des tabulations, marges gauches et droites, mais aussi, un message pour le système lui indiquant une modification de ces valeurs.

g . . . t . . . t t d

fig. 3.3 : Exemple de ligne de présentation

d) Menu

Ce type de langage est très peu utilisé ici. Il existe 2 menus (cfr. fig. 3.3) qui sont utilisés comme moyen pour saisir rapidement certaines données d'un document, à savoir son type et son état.

Rappel: type = lettre, lettre standard, formulaire,
quelconque
état = révisable, revu, signé, classé.

lettre
lettre standard
formulaire
quelconque

révisable
revu
signé
classé

fig. 3.4. : Exemples de menus

2.2.2. LES ELEMENTS MATERIELS

a) Le périphérique d'entrée

Le plus courant et celui que nous utiliserons est le clavier sur lequel l'utilisateur enfonce et relâche des touches pour envoyer un code unique pour chaque touche vers l'ordinateur (CPU - contrôleur d'entrée/sortie).

b) Le périphérique de sortie

Le but est de permettre à l'utilisateur de voir les éléments lors du traitement et les résultats de ce traitement.

On disposera essentiellement d'un écran et d'une imprimante. En plus, sur certains claviers, l'existence de voyants lumineux signalant le positionnement des modes majuscule et numérique est d'un usage très apprécié.

Description de l'écran

L'utilisateur a, en face de lui, un écran qui est logiquement divisé en trois fenêtres :

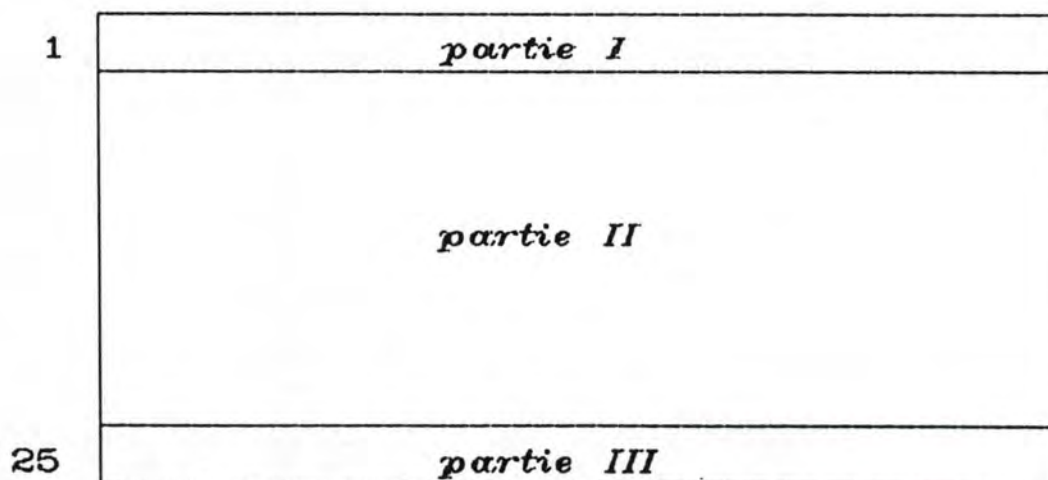


fig. 3.5 : Division de l'écran en trois fenêtres logiques

CHAPITRE III

En général, une seule de ces fenêtres est accessible à l'utilisateur : il s'agit de la fenêtre centrale. Voyons en détail le contenu de ces différentes parties.

1 Partie message (partie I sur la fig. 3.5)

Elle est constituée par la première ligne de l'écran et elle a pour but de présenter à l'utilisateur l'ensemble des messages utiles à un moment donné.

Lors de la saisie des données d'un document, la ligne de message apparaît comme suit :

(ACT COUR)	(MES TEMP)
------------	------------

- (ACT COUR) : l'action courante rapelle à tout moment ce que l'utilisateur est en train de faire (mise en page, publipostage, création document, ...).
- (MES TEMP) : zone de message temporaire. C'est là qu'apparaissent, souvent pour une durée limitée et accompagnés d'un signal sonore, les messages d'erreurs ou tout autre message utile du système.

Lors de la phase de traitement du texte, elle apparaît comme suit :

(ACT COUR)	(COL)	(LIG)	(NOM DOCUMENT)	(MES TEMP)	(CL)
------------	-------	-------	----------------	------------	------

- (ACT COUR) : cfr. supra.

- (COL) : indice de la colonne sur laquelle se trouve le curseur dans le texte.

- (LIG) : indice de la ligne sur laquelle se trouve le curseur dans le texte.

CHAPITRE III

- (NOM DOCUMENT) : nom du document dont le texte est affiché dans la partie texte de l'écran.
- (MESS TEMP) : cfr. supra.
- (CL) : mode dans lequel se trouve le clavier.
(insertion, remplacement, soulignement).

2. Partie texte (partie II sur la fig. 3.5)

Cette partie est destinée à contenir

- Lors de la saisie des données d'un document des grilles d'informations dont le traitement de texte a besoin pour
 - . sélectionner un document
 - . manipuler un document
 - . collecter les informations relatives à un document

Ces grilles d'information sont en fait les fiches à remplir et les menus sus-mentionnés (cfr. chap III 2.2.1).

Des sous-fenêtres vont apparaître sur la partie centrale de l'écran, selon le choix de l'utilisateur, afin de contenir les différentes grilles ou toute autre information nécessaire à l'exécution d'une tâche (ou pour la faciliter).

Par exemple, lors de l'effacement ou du changement de nom d'un document, lorsque le système demande, dans une grille, le nom du document à effacer ou à renommer, l'utilisateur peut faire apparaître dans une fenêtre annexe, la liste de tous ses documents.

- Lors du traitement du texte la partie texte d'un document que l'on va
 - . créer
 - . modifier
 - . mettre en page

C'est dans cette partie, essentiellement, que l'utilisateur va travailler. C'est en fait la partie qui peut être vue comme la "feuille de papier" sur laquelle il va rédiger son document : c'est sa zone de travail.

CHAPITRE III

C'est cette partie qui va permettre d'avoir une vue sur le texte selon le principe du parchemin (cfr. chap I fig. 1.1).

Dans cette zone peuvent également apparaître des sous-fenêtres qui permettent pendant un instant d'exécuter une tâche annexe.

Par exemple, lors de la création d'un texte, on peut utiliser le dictionnaire afin d'accélérer ou de faciliter l'élaboration du document. Le contenu de ce dictionnaire peut être visualisé à tout instant lors de la création. Une sous-fenêtre apparaît sur le texte créé (à un endroit qui ne gêne pas la création du texte) dans laquelle le contenu du dictionnaire peut être analysé.

CREATION DOCUMENT /col: 1/lig: 1 doc1 CL I							
<p>Lors de l'édition d'un texte quelconque, l'utilisateur peut faire appel au dictionnaire pour introduire rapidement son texte. Celui-ci apparaît dans une sous-fenêtre.</p>							
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p>DICTIONNAIRE</p> <p>Cceci est un paragraphe standard dont la référence est C</p> <p>Vvoici un second dont la référence est V</p> <p style="text-align: right;">?</p> </div>							
1		4+ ligne	5soulig	6	7dict	8tab	9enreg 0annul

3. Partie commande (partie III sur la fig. 3.5)

Elle est constituée par la dernière ligne de l'écran et elle a pour but de présenter à l'utilisateur l'ensemble des commandes qu'il peut exécuter à un moment donné.

Cette ligne de commande se présente comme suit :

jeu	n°f fct	n°f fct	n°f fct	n°f fct		n°f fct	n°f fct
-----	---------	---------	---------	---------	--	---------	---------

CHAPITRE III

où jeu : représente l'ensemble auquel les fonctions affichées (fct) appartiennent (cfr. chap III 2.2.1)

n° : est le numéro de la fonction en question au sein de l'ensemble "jeu". C'est aussi, concrètement, le numéro d'une des dix touches fonctions situées sur le clavier

fct : est une abbréviation (mnémonique) de la fonction.

Après chaque opération exécutée par l'utilisateur, cette ligne est susceptible de se modifier : elle cache les commandes non exécutables à un moment donné en n'affichant que les commandes disponibles (généralement, sous la forme d'utilisation des touches fonctions).

CHAPITRE III

Par exemple, prenons un scénario possible sur le traitement de texte et analysons la suite des écrans en faisant particulièrement attention à la dernière ligne.

Supposons que nous voulons effacer un document précédemment enregistré sur une disquette.

Nous nous trouvons dans l'état initial suivant :

TRAITEMENT DE TEXTE							
0	1créat	2modif	3mis pag	4impres	5manip		0annul

Nous sélectionnons la fonction 5, car nous voulons manipuler un document (cfr. description chap III 2.1.1).

L'écran ci-dessous apparaît :

MANIPULATION DE DOCUMENT							
4		4adresse	5repert	6efface	7renomme	8rec doc	.. 0annul

CHAPITRE III

Nous sélectionnons ensuite la fonction 6 pour effacer notre document.

L'écran ci-dessous apparaît :

<div>Nom du document :</div>			
4		5repert	0annul

A présent, nous pouvons écrire le nom du document que nous voulons effacer ou alors, nous pouvons sélectionner, par exemple, la fonction 5 afin d'afficher sur notre écran le répertoire des documents (afin de se rappeler, par exemple, le nom ou l'orthographe de celui-ci).

Nous obtenons l'écran suivant :

<div>Nom du document :</div>				<div>nom doc date auteur nom doc date auteur . . .</div>		
4						0annul

CHAPITRE III

Nous pouvons alors écrire correctement le nom du document ; mais, nous pouvons toujours, à ce stade, annuler notre requête grâce à la fonction 10 (la seule disponible à cet instant).

En général, la seule commande suffit et est immédiate. Pour certaines qui s'exécutent avec paramètres, ceux-ci sont demandés à l'utilisateur dans cette ligne de commande.

IV. IMPLEMENTATION DU TRAITEMENT DE TEXTE (Hammer, 9)

Ce quatrième chapitre expose d'abord la manière selon laquelle les données sont représentées en mémoire centrale et sur les supports de mémorisation.

Ensuite, nous présenterons l'architecture du système.

Enfin, les différents modules seront spécifiés selon le format suivant : les entrées, les sorties, les pré/post conditions, la fonction, ainsi qu'éventuellement une structure première de l'algorithme.

CHAPITRE IV

1. DESCRIPTION DES FICHIERS ET UTILISATION

Au dessus du modèle conceptuel du document, il existe une structure physique plus générale reprenant l'ensemble des composants du traitement de texte c'est-à-dire l'ensemble des fichiers nécessaires à son bon fonctionnement.



L'implémentation du document rédigé et géré par l'utilisateur est réalisée à l'aide de deux fichiers distincts : ce sont les fichiers "document" et "données document". Lorsque l'utilisateur crée un document, il crée donc ces deux fichiers de telle sorte que dans le système, à chaque fichier "document" correspond toujours un et un seul fichier "données document".

1.1. Le fichier "document"

1.1.1. DESCRIPTION

Le fichier "document" est destiné à contenir la partie texte de la structure logique.

Ce fichier est un fichier texte (text file en Pascal) c'est-à-dire un fichier dont les composants sont des caractères. Les caractères permis sont tout ceux qui sont repris dans le tableau des codes ASCII à l'exception des 31 premiers qui sont réservés pour le système.

En fait, seulement quelques-uns sont utilisés par le système, à savoir les caractères dont le code ASCII suit :

- 9 : (CTRL-I) caractère spécial représentant le début d'une zone de texte en interligne réduit (bloc texte) ; la zone reste à interligne réduit jusqu'au moment où un second caractère de ce type est rencontré.

CHAPITRE IV

- 13 : (CR) caractère spécial représentant la fin d'une ligne.
- 19 : (CTRL-S) caractère spécial représentant le début d'une zone de texte souligné ; la zone reste soulignée jusqu'au moment où l'on rencontre à nouveau ce caractère.
- 26 : (CTRL-Z) caractère spécial représentant la fin d'un texte.

1.1.2. STRUCTURE DE MANIPULATION

Pour des raisons d'efficacité, on ne va pas travailler sur le fichier texte lui-même ; celui-ci n'est utilisé que pour le stockage. On va plutôt copier ce fichier dans un grand tableau nommé BUFFER ; celui-ci, à la fin du traitement, est recopié dans le fichier.

a) Buffer

La structure du tableau "BUFFER" est celle d'un "array" en Pascal. Il est défini sur - 350 lignes
- 81 colonnes.

Le nombre de colonnes est en fait le nombre de colonnes de l'écran auquel nous avons ajouté une colonne supplémentaire. Elle est nécessaire car toute ligne est terminée par un caractère (invisible à l'écran) de fin de ligne qui doit être mémorisé dans le tableau ; il est placé après le dernier caractère visible à l'écran. Or une ligne pleine présente à l'écran 80 caractères visibles.

[illegible]

ne peut contenir que des
caractères "fl" de fin de
ligne

CHAPITRE IV

b) Carcar

Au début du traitement d'un document, lors de la copie du fichier dans BUFFER, on va également créé un second tableau structurellement identique au premier. Ce tableau ("CARCAR") est créé à partir des caractères spéciaux (code ASCII 9 et 19) contenus éventuellement dans le fichier "document" et complété par des commandes de l'utilisateur (pose de marques, ...).

C'est en fait une matrice d'entiers dont chaque case est associée biunivoquement au caractère de BUFFER ayant les mêmes coordonnées. Cette matrice codée se lit de la façon suivante :

- code 1 : caractère souligné
- 2 : caractère marqué comme début ou fin de bloc logique (affiché en vidéo inverse et clignotant)
- 3 : ce code ne peut se trouver que dans la colonne 81 de CARCAR et signifie que cette ligne est la dernière ligne d'une page
- 4 : caractère marqué comme début ou fin de bloc texte (affiché en clignotant)
- 0 : autre

c) F début et f fin

Etant donné la taille fixe imposée par le tableau pour créé un document, il faudrait implémenter, afin de pouvoir traiter des documents de plus de 350 lignes, un système de vidage du tableau dans des fichiers de travail.

On notera que la manipulation de tel texte entame en partie les performances du système mais on se rappellera que le traitement de texte réalisé est destiné à des cadres supérieurs et que la longueur des documents qu'ils seront amenés à taper eux-mêmes n'est probablement pas importante.

fichier "document" = f_début + buffer = f_fin

Pour ces deux raisons, nous n'implémenterons pas concrètement ce système de vidage dans ce travail. Nous proposerons toutefois, afin d'être complet, une solution théorique à ce problème (cfr. chap IV 1.1.2 d).

d) Vue générale du document

Globalement, le document apparaît comme suit :

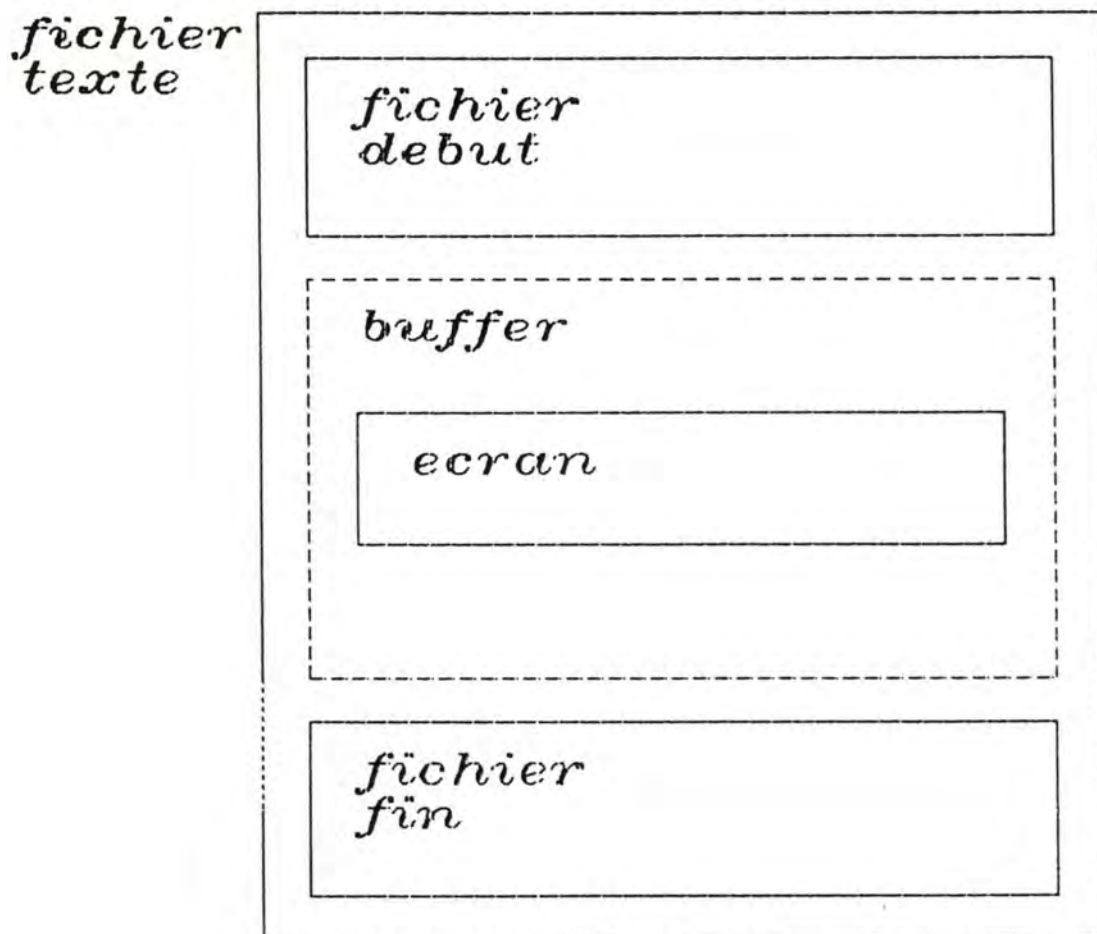


fig. 4.1 : Vue générale d'un document

- l'écran peut être vu comme une fenêtre qui voyage verticalement sur le tableau BUFFER afin de pouvoir le visualiser entièrement.
On a une variable de positionnement de l'écran qui indique l'indice dans le tableau de la première ligne de l'écran.
- le tableau peut aussi être vu comme une fenêtre qui voyage verticalement sur le fichier document.
On a une variable fin-f-début qui est le numéro de la dernière ligne enregistrée dans f_début.
- ainsi le numéro d'une ligne dans tableau
= indice fin f_début + n° ligne dans tableau

1.2. Le fichier "données document"

Le fichier "données document" est destiné à contenir les parties "type document", "entête", "clôture" de la structure logique et la partie "format" de la structure physique.

Ce fichier est en fait un seul enregistrement ("record" en Pascal) c  d un ensemble d'informations de types diff  rents, mais qualifiant un m  me objet, en l'occurrence un document.

Cet ensemble d'informations se pr  sente comme suit :

```

document = record
    typdoc : c_type;
    entete : c_entete;
    cloture : c_cloture;
    format : c_format;
end;

o   c_type = (lettre, lettre standard, formulaire, document
quelconque)

c_entete = record
    nom           : string10;
    sujet         : suj;
    expedition    : string10;
    dest          : t_dest;
    date_emission : string10;
end;

c_cloture = record
    date_reception : string10;
    etat           : r  visable, revu, envoy  ,
                  ou class  
end;

c_format = record
    mg, md, bm, tm, lp : integer;
    ir                  : char;
end;
    
```

La r  alisation du traitement de texte sp  cifi   implique encore l'impl  mentation de trois autres fichiers.

Le premier, appel   "tabulation", est enti  rement g  r   par le syst  me et l'utilisateur n'y a pas acc  s.

CHAPITRE IV

1.3. Le fichier "tabulation"

Le fichier "tabulation" est destiné à contenir les tabulations par défaut qui seront utilisées lors de la création d'un document car celles-ci varient suivant le type de document traité.

C'est un fichier contenant une séquence de quatre enregistrements ("record Pascal) qui se présente comme suit :

```
tabulation = record
    typtab : (lettre, lettre standard, formulaire,
              document quelconque)
    tabul : array °1..80$ of char;
end;
```

	1	4	8	24	25	50	77	80
lettre		mg			t	t	md	
let st		mg			t	t	md	
formul	mg		t	t				md
doc qcq	mg		t	t				md

Il y a enfin deux fichiers créés et gérés par l'utilisateur. Ces fichiers ne sont pas obligatoires mais peuvent être très utiles pour l'utilisation efficace du traitement de texte.

1.4. Le fichier "liste d'adresses"

Le fichier "liste d'adresses" peut exister dans le système en exemplaires multiples, chacun d'eux contenant des listes d'adresses : ces adresses seront les destinataires des documents établis et pourront être classées par sous-groupes dans les différents fichiers.

Les listes d'adresses seront utilisées à deux niveaux :

- lors du publi-postage

un document sera imprimé en plusieurs exemplaires ; chacun d'eux est adressé à un destinataire dont l'adresse se trouve dans la liste.

CHAPITRE IV

- lors de la saisie des données d'une lettre

au cas où l'utilisateur ne mentionne que le nom du destinataire de sa lettre et si l'adresse de celui-ci est pré-enregistrée dans un quelconque des fichiers "liste d'adresses", alors la saisie de l'adresse du destinataire est complétée par les données du fichier.

au cas où il y aurait deux destinataires ayant même nom

dans un même fichier, on donnera toujours priorité à la première occurrence

dans des fichiers différents, la priorité est accordée suivant l'ordre d'apparition de ces fichiers dans le répertoire.

Ces fichiers auront obligatoirement, dans le nom qui les identifie, une extension sous la forme ".ADR".

Il aura donc la forme suivante :

(nom de fichier) = (suite de 6 car).adr

Un tel fichier contient une séquence de longueur quelconque d'enregistrements ("record" en Pascal) qui se présentent comme suit :

```
t_dest = record
    nom      : string30;
    qual     : string30;
    adr_n    : string30;
    cp_loc   : string30;
end;
```

Exemple :

DUPONT	Directeur	rue Blondeau	40	5000 NAMUR
LEPONT	Manager	rue de Bruxelles	100	5000 NAMUR
LEDUR		rue des Bambous	45	1000 BRUXELLES

1.5. Le fichier "dictionnaire"

S'il existe, il est unique dans le système.

C'est un ensemble de mots, phrases, paragraphes - c-à-d une suite quelconque de caractères que l'utilisateur a voulu mémoriser car elle revêt pour lui une importance particulière (par exemple : il veut souvent y faire référence). A cette suite est associée de façon unique un identifiant qui est la référence, choisie par l'utilisateur, de la suite mémorisée.

C'est un fichier texte (text file en Pascal) comme le fichier "document" mais il faut donner au texte une structure bien spécifique :

```
id texte !  
id texte !
```

```
.  
.  
.
```

où id est l'identifiant

id = le premier caractère du fichier "dictionnaire"
ou n'importe quel autre caractère différent de
"cr" qui suit immédiatement le caractère de
fin de définition "!".

Le fichier "dictionnaire" a obligatoirement le nom "DICT".

Si deux identifiants sont les mêmes, seule, la première occurrence qui apparaît dans le dictionnaire sera utilisée.

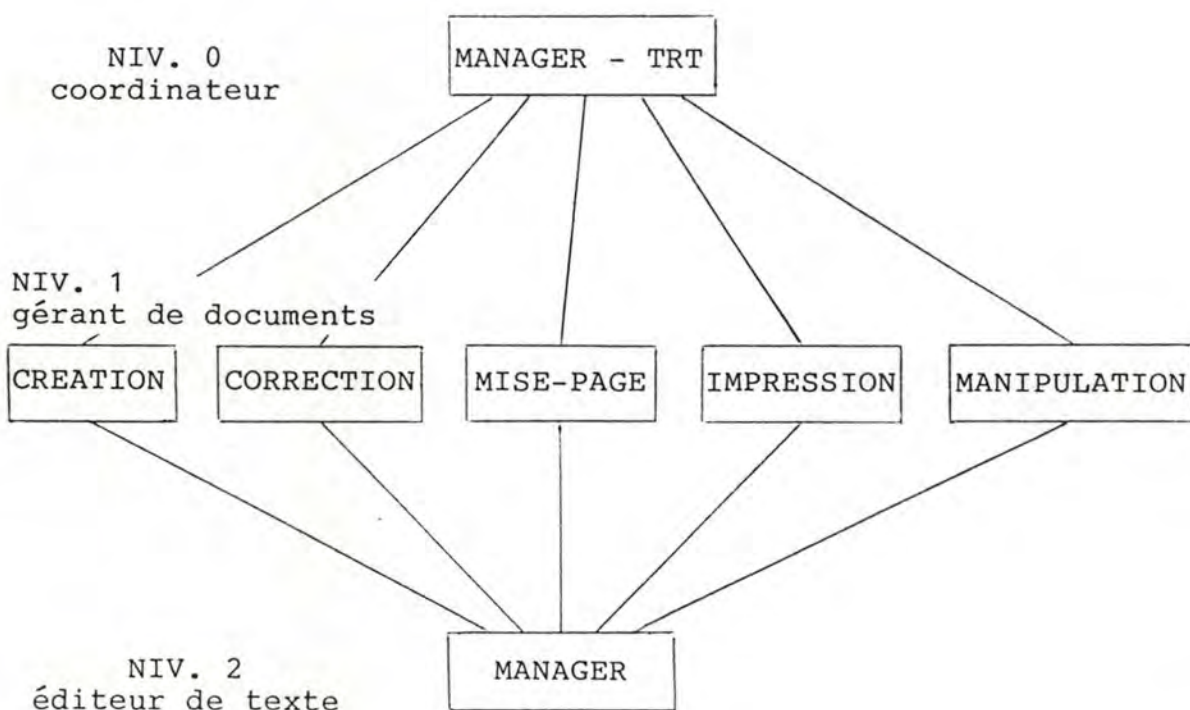
2. L'ARCHITECTURE DU SYSTEME

2.1. Introduction

L'implémentation du traitement de texte a été réalisée sur l'Olivetti M-24 dont le système d'exploitation est le MS-DOS. Le langage de programmation qui a été choisi est le Turbo Pascal : le Pascal pour la structuration aisée des données et leur manipulation sur fichier et dans les tableaux ; le Turbo pour bénéficier de certaines primitives de gestion d'écran.

Dans cette partie, nous allons présenter une description sommaire de l'architecture du système.

L'implémentation du système est réalisée par un ensemble de modules qui peuvent être organisés en une hiérarchie d'appels à trois niveaux : au premier niveau (le plus haut), nous avons un module de coordination ; au deuxième, un ensemble de modules pour la gestion de documents ; enfin, au troisième niveau, les modules pour l'édition de texte :



L'enchaînement des modules est réalisé par les relations "Appelle" et "Est appelé par".

2.2. Analyse de la découpe modulaire

Nous allons ici présenter les différents arbres modulaires qui existent à chacun des niveaux d'implémentation.

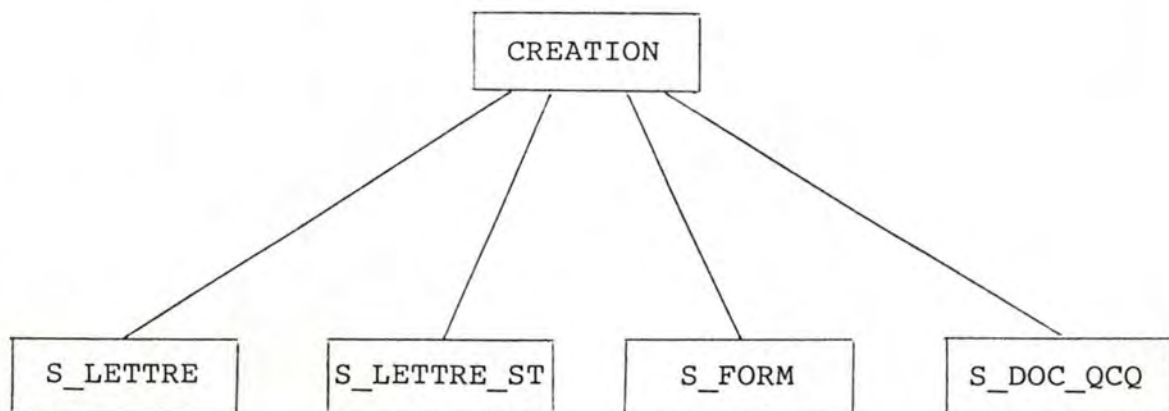
2.2.1. Niveau 0

Il est constitué par un seul module qui est le coordinateur. Il a pour but de gérer l'orientation de l'utilisateur dans les modules de niveau 1.

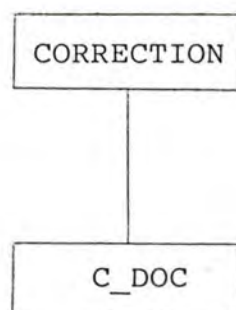
2.2.2. Niveau 1

A ce niveau, nous avons cinq classes de modules à partir desquelles on peut constituer de nouveaux arbres modulaires. Ils ont pour but de saisir l'information nécessaire à la gestion des documents et de gérer les documents eux-mêmes.

Classe 1 : la création d'un document

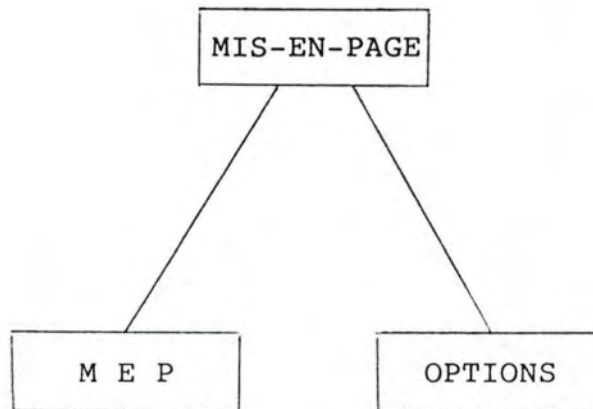


Classe 2 : la correction ou modification d'un document

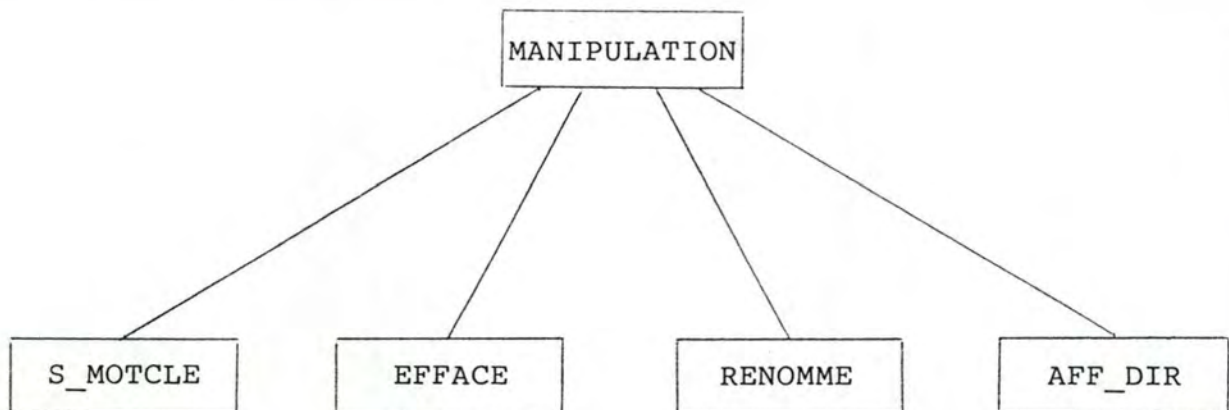


CHAPITRE IV

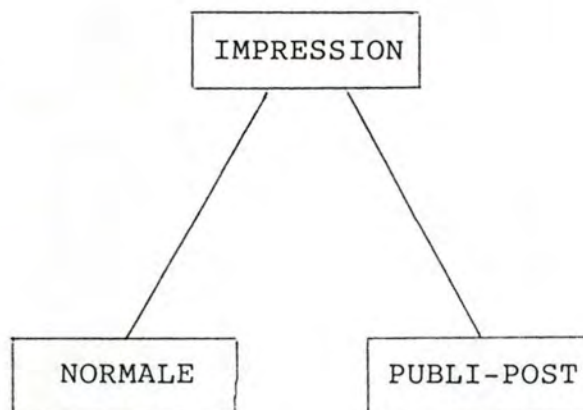
Classe 3 : la mise en page d'un document



Classe 4 : la manipulation de documents



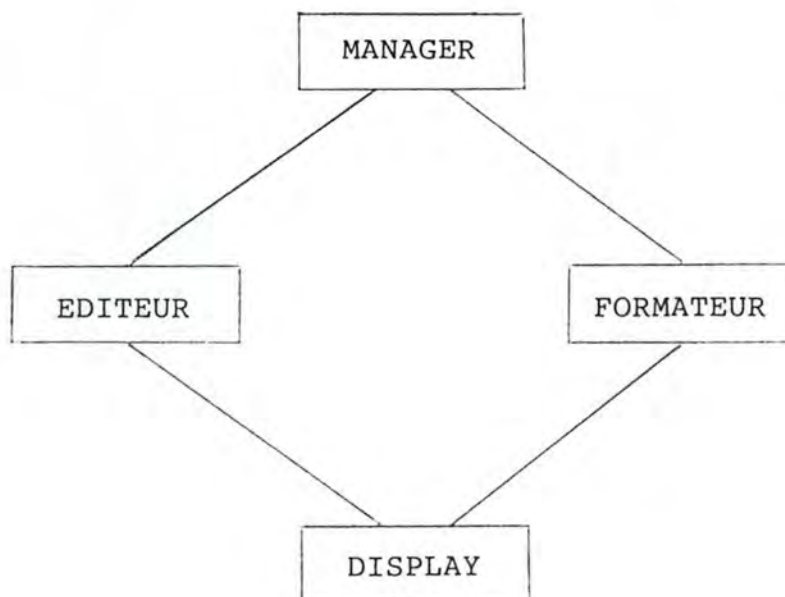
Classe 5 : l'impression d'un document



CHAPITRE IV

2.2.2. Niveau 2

Dans ce dernier niveau, nous avons un seul module à partir duquel on peut constituer l'arbre modulaire ci-dessous. Il a pour but de gérer tout ce qui concerne l'édition d'un document.



Le "manager" est responsable de l'analyse des touches frappées sur le clavier par l'utilisateur, de leur interprétation et de l'appel des modules internes appropriés pour réaliser l'action désirée.

L'"éditeur" est responsable de la réalisation des changements (création, modification) dans le texte d'un document (dans BUFFER et CARCAR).

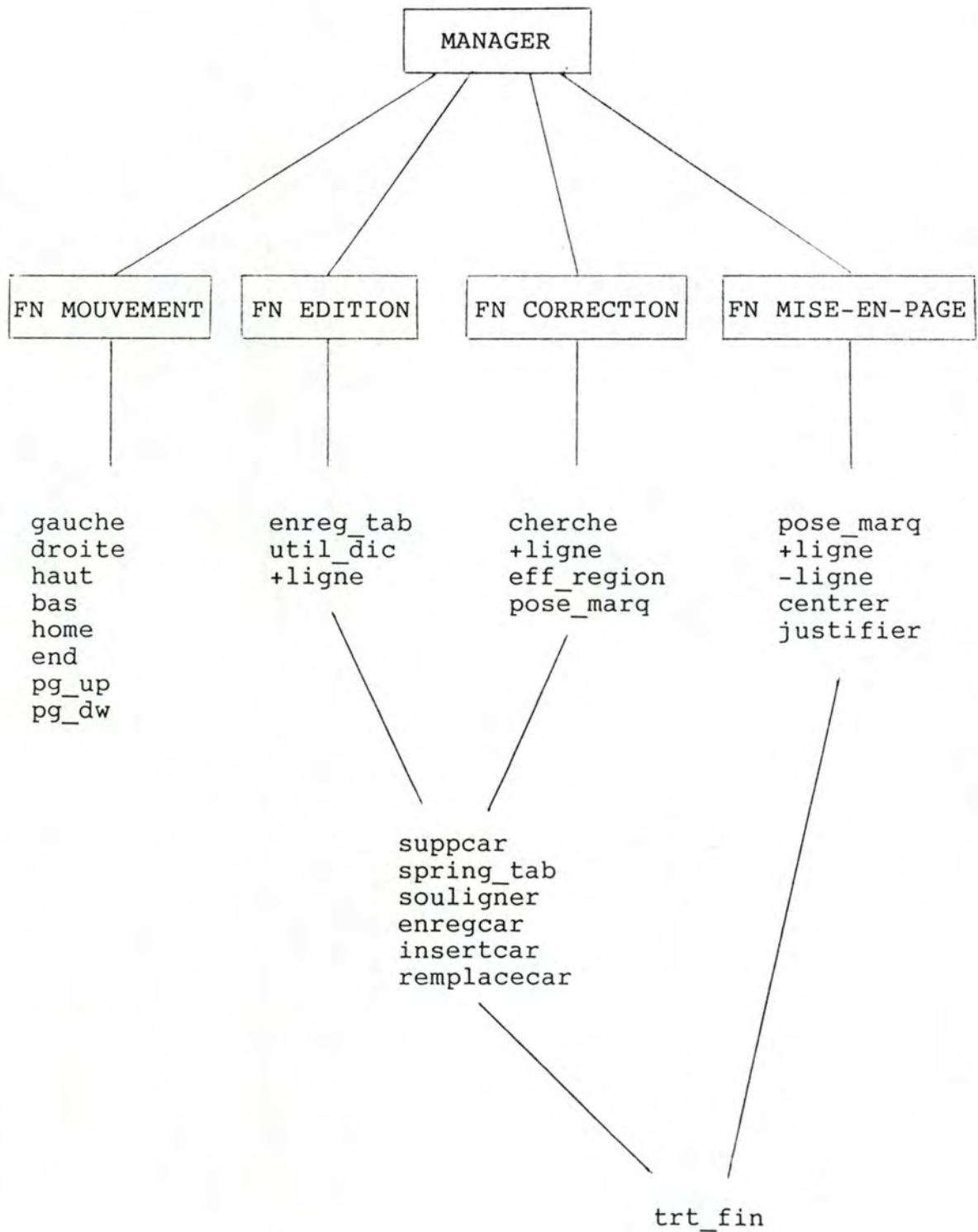
Le "formateur" permet une remise en page de tout un document.

Le "display" est invoqué après l'"éditeur" ou le "formateur" afin de mettre à jour l'écran pour refléter l'état actuel du document. C'est le gestionnaire d'écran proprement dit.

On a donc toujours le modèle de base suivant : le "manager" appelle soit l'"éditeur", soit le "formateur", et ensuite, le module "display" est appelé.

CHAPITRE IV

On peut encore détailler un peu cette structure modulaire pour faire apparaître l'ensemble des fonctions disponibles à ce niveau.



CHAPITRE IV

3. SPECIFICATION DES DIFFERENTS MODULES

Pour la spécification des modules, nous respecterons la découpe modulaire exposée dans la partie précédente

Afin de comprendre ces spécifications, il convient dès à présent, de préciser certains termes que nous utiliserons souvent dans cette partie ; on n'hésitera pas cependant à se référer également au lexique en fin de travail.

On parlera souvent de différents modes. Il faut savoir que l'on distingue :

- le mode soulignement : s'il est actif, tout caractère introduit est souligné
sinon, il ne l'est pas.
- le mode de remplacement :
s'il est actif, le caractère tapé remplace le caractère courant
sinon on est en mode insertion où le caractère tapé est inséré
devant le caractère courant.

Le caractère courant est celui qui se trouve à la position courante c'est-à-dire la position du curseur.

L'indicateur de positionnement du tableau est un nombre entier qui représente le numéro de la première ligne de l'écran dans BUFFER.

L'indicateur de reformatage est une valeur booléenne qui signale si un texte doit être reformaté ou non (càd si sa marge gauche ou droite a été modifiée).

CHAPITRE IV

3.1. Spécification des modules de niveau 0

Module MANAGER TRT

input : caractère frappé par l'utilisateur

appelle : PRESENTATION, AFF_FCT, CREATION, CORRECTION,
MIS_EN_PAGE, IMPRESSION, MANIPULATION, MANAGER

fonction : suivant le caractère donné en entrée, le module initialise correctement les données (buffer, carcar, top_tab, n° dernière ligne, fin de page, caractère de fin de ligne et l'indicateur de soulignement) et sélectionne le module de niveau inférieur désiré.

Module INIT EXIST

input : fichier texte
marge gauche, n° de la dernière ligne

output : position courante maj
buffer, carcar maj
n° de la dernière ligne maj

appelé par : S_LET_ST, MEP, CORRECTION

fonction :

lit le contenu du fichier texte et initialise BUFFER et CARCAR
càd :

1. place les caractères dans BUFFER de telle sorte que
 - un caractère de fin de ligne (ASCII 13) est enregistré dans le tableau et indique le passage à la ligne suivante
 - en fin de fichier texte, un caractère de fin de texte (ASCII 26) est placé dans BUFFER
 - les autres caractères sont simplement placés dans BUFFER
2. maj le tableau CARCAR en enregistrant les caractères soulignés (code 1) et les caractères de début de bloc texte (code 4).

Module INIT NONEXIST

input : marge gauche

output : buffer et position courante maj

appelé par : S_LETTRE, S_FORM, S_DOC_QCQ

CHAPITRE IV

fonction : initialise BUFFER quand il n'y a pas de fichier texte.
Le caractère de fin de texte (ASCII 26) est placé dans
BUFFER.

Module TRT FIN1

input : données document ('r_document')
n° de la dernière ligne
marge gauche et buffer

ouput : fichier texte

appelé par : MANAGER

fonction : enregistre le contenu de BUFFER et de CARCAR dans le
fichier texte de sorte que :

lors de l'enregistrement du caractère (i,j) de BUFFER

- si le caractère (i,j) de CARCAR = 4
alors le caractère spécial (ASCII 9) est enregistré
(début de bloc texte)
- si le caractère (i,j) de CARCAR = 1
alors le caractère spécial (ASCII 19) est enregistré
(début de zone soulignée)
- si le caractère (i,j) de CARCAR = 0 et que l'on ait jusqu'à
présent enregistré un nombre impair de caractère spécial (ASCII
19)
alors un nouveau caractère spécial (ASCII 19) est enregistré
(fin de zone soulignée)
- sinon le caractère (i,j) de BUFFER est enregistré dans le
fichier texte.

3.2. Spécification des modules de niveau 1

Module CREATION

input : fichier tabulation
données concernant un document ('r_document.typdoc')

output : ligne de présentation maj
mode clavier, remplacement maj
écran maj

appelle : MESSACT, AFF_FCT, FRAME, CHOIX_TAB,
S_LETTRE, S_LET_ST, S_FORM, S_DOC_QCQ

appelé par : MANAGER_TRT

fonction : suivant le choix de l'utilisateur, sélectionne le type de document à créer, initialise la ligne de présentation et le mode du clavier et de remplacement.

Module S LETTRE.

output : maj données concernant le type de document lettre (cfr. supra - type de document)

appelle : REVERSE, NORMAL, S_NOM, INIT_R_DOCUMENT, INIT_NONEXIST, LIRESTR, S_DATE, S_DEST, AFF_FCT, FRAME

appelé par : CREATION

fonction : établissement progressif d'une grille de saisie des données concernant le type de document 'lettre' et saisie de :
- nom de la lettre
- mots-clés
- expéditeur
- date d'émission
- destinataire

Module S FORM.

output : maj des données concernant le type de document formulaire (cfr. supra - type de document)

appelle : AFF_FCT, FRAME, REVERSE, NORMAL, S_NOM, INIT_R_DOCUMENT, INIT_NONEXIST

appelé par : CREATION

CHAPITRE IV

fonction : établissement progressif d'une grille de saisie des données concernant le type de document 'formulaire' et saisie de

- nom du formulaire
- date d'émission

Module S LET ST.

output : maj des données concernant le type de document lettre standard (cfr. supra - type de document)

appelle : FRAME, AFF_FCT, REVERSE, NORMAL
S_NOM, INIT_R_DOCUMENT, INIT_EXIST, RESTORE_SCREEN

appelé par : CREATION

fonction : établissement progressif d'une grille de saisie des données concernant le type de document 'lettre standard'.

- affiche le formulaire à remplir sur l'écran
- saisie du nom de la lettre standard
- saisie de toutes les données relatives à une lettre normale

(cfr. supra - S_LETTRE).

Module S DOC QCQ.

output : maj des données concernant le type de document quelconque
(cfr. supra - type de document)

appelle : FRAME, AFF_FCT, REVERSE, NORMAL,
S_NOM, INIT_R_DOCUMENT, INIT_NONEXIST

appelé par : CREATION

fonction : établissement progressif d'une grille de saisie des données concernant le type de document 'quelconque'. -
affiche le formulaire à remplir sur l'écran

- saisie du nom du document quelconque

Module CORRECTION

output : écran maj
mode du clavier et de remplacement maj

appelle : MESSACT, AFF_FCT, FRAME, C_DOC, INIT_EXIST,
RESTORE_SCREEN

appelé par : MANAGER_TRT

fonction : initialise la saisie, l'écran et les modes clavier et remplacement.

CHAPITRE IV

Module C DOC.

input : données intrinsèques d'un document

output : données intrinsèques d'un document maj

appelle : S_NOM, INIT_R_DOCUMENT, CHOIX_TAB, S_DATE

appelé par : CORRECTION

fonction :

- à partir du nom du document, affiche les valeurs actuelles (ou par défaut) des données intrinsèques du document
- permet de modifier ces valeurs, à savoir :
 - . l'état d'un document non formulaire
 - . date de réception
- permet d'accepter ces nouvelles valeurs et l'ensemble des valeurs maj

Module MIS EN PAGE

input : nom du document
caractère tapé par l'utilisateur

output : écran maj

appelle : AFF_FCT, MESSACT, FRAME, SAISIE_NOM, INIT_R_DOCUMENT, MEP, OPTIONS

appelé par : MANAGER_TRT

fonction : à partir du nom d'un document, initialise la mise en page d'un document

Module MEP.

input : marge haut, marge gauche
buffer, carcar
numéro de la dernière ligne
indicateur de reformatage
indice de positionnement du tableau
position courant et longueur de page

output : buffer, écran et position courante maj
numéro de la dernière ligne maj

appelle : INIT_VEDETTE, REFORMAT, INIT_EXIST, TMBM_INSERT, RESTORE_SCREEN, AFF_FCT, MESSPOS

appelé par : MIS_EN_PAGE, PUBLI_POST, NOTM_IMPRES

CHAPITRE IV

fonction :

- introduit une vedette dans BUFFER s'il s'agit d'une lettre
càd copie dans BUFFER, à paritr de la ligne tm+2, les
(4) données concernant le destinataire
- découpe BUFFER en page càd introduit les mages haut et bas
suivant la longueur de page.

Module OPTIONS.

input : marge gauche, droite, haut, bas
longueur de page, indicateur de réduction d'interligne
caractères frappés par l'utilisateur

output : marge gauche, droite, haut, bas maj
longueur de page, indicateur de réduction d'interligne
maj
écran maj
indicateur de reformatage maj

appelle : LIRESTR

appelé par : MIS_EN_PAGE

fonction : permet de visualiser les différentes options de
formatage dans une fiche et de modifier ces options
si nécessaire.

Module MANIPULATION

input : caractère tapé par l'utilisateur

output : écran maj

appelle : FRAME, AFF_FCT, AFF_DIR, EFFACE, RENOMME, CH_DOC

appelé par : MANAGER_TRT

fonction : initialise l'écran afin de proposer toutes les
manipulations possibles à l'utilisateur et gère son
choix.

Module EFFACE.

input : répertoire

output : répertoire maj
message sur écran

appelle: AFF_FCT, MESSACT, FRAME,
AFF_DIR, LIRESTR, MAJ_DIR

CHAPITRE IV

appelé par : MANIPULATION

fonction :

- à partir du nom - que l'on peut retrouver par l'affichage simultané du répertoire -, le document désigné est supprimé du support externe de mémorisation.
- envoie un message
- le répertoire est maj par une suppression logique du document c-à-d que le nom du document est mis à blanc
- éventuellement, ré-organisation du répertoire avec suppression physique des documents logiquement supprimés afin d'éviter la gestion d'un fichier en continuelle expansion

Module RENOMME.

input : répertoire

output : répertoire maj

appelle : AFF_FCT, MESSACT, FRAME, AFF_DIR, LIRESTR, S_NOM

appelé par : MANIPULATION

fonction :

- à partir du nom - que l'on peut retrouver par l'affichage simultané du répertoire -, le document désigné change de nom sur le support externe de mémorisation.
- envoie un message
- le répertoire est maj par une modification du nom du document

Module IMPRESSION

input : caractère tapé par l'utilisateur

output : écran maj

appelle : FRAME, AFF_FCT, NORM_IMPRES, PUBLI_POST

appelé par : MANAGER_TRT

fonction : initialise l'écran afin de proposer toutes les impressions possibles à l'utilisateur et gère son choix.

CHAPITRE IV

Module PUBLI POST

input : nom de la lettre standard
 nom du fichier d'adresses

output : écran maj
 états imprimés des lettres publi-postées

appelle : BRANCHER, MEP, MESSDEBUT, PRINT_BUFFER

appelé par : IMPRESSION

fonction : à partir du nom d'une lettre standard et d'un fichier d'adresses, imprime autant de lettres personnalisées qu'il n'y a d'adresses dans le fichier nommé.

Module NORM IMPRES

input : nom du document

output : état imprimé du document
 écran maj

appelle : BRANCHER, MEP, MESSDEBUT, PRINT_BUFFER

appelé par : IMPRESSION

fonction : à partir du nom d'un document, imprime le document.

CHAPITRE IV

3.3. Spécification des modules de niveau 2

3.3.1. MODULES DU MANAGER

Module MANAGER

input : un caractère tapé par l'utilisateur
buffer, code, indicateur de soulignement

appelle : MESSDEBUT, VALIDATION,
GAUCHE, DROITE, BAS, HAUT, HOME, G_END, PG_DW, PG_UP,
SUPPCAR, REPLMOD, SOULIGNER, TRTFIN1, TRTFIN2,
ENREG_TAB, UTIL_DICT, INSERT, CHERCHE, EFF_REGION,
POSE_MARQ, MOINS_LIGNE, Z_CENTRER, Z_JUSTIFIER,
JUMPHOLE, SPRINGTAB, SOULIG1,
ENREGCAR, REMPLACECAR, INSERTCAR

appelé par : MANAGER_TRT

fonction : sélectionne le module de niveau inférieur selon le caractère d'entrée validé.

Module CASSDAT.

input : date 'd' sous forme d'une chaîne de 10 caractères

output : date sous la forme de trois entiers
dd : représentant le jour
dm : représentant le mois
dy : représentant l'année

pré : la chaîne de 10 caractères doit avoir le format suivant
caractère 1-2 représente le jour
" 4-5 représente le mois
" 7-10 représente l'année

appelle : STRINGINT

appelé par : VERIFDAT

fonction : décompose une suite de caractères représentant une date en trois entiers représentant le jour, le mois, l'année.

Module VERIFDAT.

input : date 'd' sous la forme d'une chaîne de 10 caractères

output : booléen 'datecor' représentant le fait qu'une date est valide (cfr. fonction infra) ou pas

appelé par : S_DATE

CHAPITRE IV

fonction : vérifie si une date est valide c'est-à-dire si elle correspond à notre calendrier habituel à savoir 12 mois de 31/30/29/28 jours avec une année d'au maximum 2099.

Module DATE

output : date sous forme d'une suite de 10 caractères

appelle : MSDOS

c'est un appel système qui par l'intermédiaire d'un enregistrement au format bien spécifique ('repack') permet d'obtenir des informations gérées par le système MSDOS ; en l'occurrence, ici, la date connue dans le système.

appelé par : S_DATE

fonction : obtenir, par l'intermédiaire d'un appel système, la date connue dans le système.

Module REVERSE.

appelé par : S_LETTRE, S_FORM, S_LET_ST, S_DOC_QCQ

fonction : permet un changement de couleur lors des écritures. toute écriture future sur l'écran sera en vidéo inverse c'est-à-dire le texte en couleur sombre et le fond en couleur claire (sur les écrans monochromes).

Module REVERSEBLINK.

appelé par : CHOIX_TAB, RESTORE_SCREEN, WRSCREEN, POSE_MARQ

fonction : permet un changement de couleur lors des écritures. toute écriture future sur l'écran sera en vidéo inverse mais en plus le texte en couleur sombre apparaîtra en clignotant.

Module BLINK.

appelé par : RESTORE_SCREEN, POSE_MARQ

fonction : permet un changement de couleur lors des écritures. toute écriture future sur l'écran apparaîtra en clignotant.

CHAPITRE IV

Module NORMAL.

appelé par : S_LETTRE, S_FORM, S_LET_ST, S_DOC_QCQ, CHOIX_TAB,
RESTORE_SCREEN, WRSCREEN, POSE_MARQ

fonction : permet de rétablir une écriture dite normale (texte clair sur fond sombre).

Module FRAME.

input : coordonnées de 2 points qui représentent respectivement le coin supérieur gauche et le coin inférieur droit d'un cadre.

ouput : écran maj

pré: les coordonnées doivent être comprises dans les coordonnées de l'écran, à savoir :
0 <= x <= 79
0 <= y <= 24

appelé par : S_LETTRE, S_FORM, S_LET_ST, S_DOC_QCQ, C_DON_DOC
EFFACE, RENOMME, CRÉATION, CORRECTION, MIS_EN_PAGE,
MANIPULATION, IMPRESSION

fonction : trace sur l'écran un cadre dont on a donné les coordonnées du coin supérieur gauche et inférieur droit.

Module EXIST.

input : nom d'un fichier texte

output : booléan qui est vrai si le fichier en question existe faux sinon.

fonction : vérifie l'existence d'un fichier de type texte ('text') et initialise ce fichier pour lecture/écriture s'il existe.

Module INIT R DOCUMENT.

input : nom d'un fichier document

output : données du document initialisées

appelé par : S_LETTRE, S_FORM, S_LET_ST, S_DOC_QCQ, C_DOC,
MIS_EN_PAGE

CHAPITRE IV

fonction :

initialise les différentes données du document (cfr. supra), à savoir :

- les mots-clés : 4 suites de caractères initialisées à blanc.
- l'expéditeur : 1 suite de caractères initialisée à blanc.
- la date d'émission : 1 suite de caractères initialisée à blanc.
- l'adresse : 4 suites de caractères initialisées à blanc.
- l'état : initialisé avec la valeur possible "révisable".
- la date de réception : 1 suite de caractères initialisée à blanc.
- le format
 - . marge droite = 80
 - . marge gauche = 1
 - . marge haut = 3
 - . marge bas = 3
 - . ligne(s)/page = 66
 - . interligne réduit = non

Module LIRESTR.

input : suite de caractères lue (suite vide ou d'un élément)
longueur maximale de la suite de caractères à lire

output : suite de caractères lue

pré : longueur maximale ≤ 255

appelé par : S_NOM, S_LETTE, C_DON_DOC, EFFACE, RENOMME,
S_MOTCLE, OPTIONS

fonction :

- affiche sur l'écran une suite de points ('.') de longueur égale à la longueur maximale de la suite à lire.
- lit une suite de caractères de telle façon que
 - . si le caractère lu est un caractère de fin de ligne (code ASCII 13), alors la lecture est terminée et la suite à lire est la suite lue jusqu'à présent.
 - . si le caractère lu est le caractère de retour en arrière (code ASCII 8), alors le dernier caractère lu est effacé et remplacé par un point sur l'écran ; il est également supprimé de la suite.

CHAPITRE IV

- . si le caractère lu est la touche fonction 10, alors la lecture est annulée et la suite à lire est remise à blanc.
- . s'il s'agit d'une autre touche fonction, il y a émission d'un signal sonore sans autres effets.
- . sinon, alors la suite lue ('str') est maj par l'adjonction du dernier caractère lu (carecr) pour former une nouvelle suite ('str').

Module S DATE.

input : suite vide ou suite de 10 caractères représentant une date

output : suite de 10 caractères représentant une date

appelé par : S_LETTRE, C_DOC

pré : l'abscisse de la position courante doit être ≤ 69

fonction :

- affiche une date (celle relative au document traité)
- si la date n'est pas acceptée (pas de caractère d'acceptation), alors la date affichée est effacée et une nouvelle date peut être saisie suivant le format d'une date valide (cfr. supra - validation date).

Module S DEST.

input : coordonnées du coin supérieur gauche d'un formulaire représentant les données du destinataire.

output : maj données concernant le destinataire d'un document

appelé par : S_LETTRE

fonction : saisie des données concernant le destinataire d'un document à savoir

- nom : suite de 30 caractères quelconques
- qualité : suite de 30 caractères quelconques
- adresse/numero : suite de 30 caractères quelconques
- code postal:localité : suite de 30 caractères quelconques

la saisie peut s'effectuer au terminal ou dans un fichier d'adresses pré-enregistré

CHAPITRE IV

Module S NOM.

input : coordonnée représentant l'endroit sur l'écran où la saisie a lieu un indicateur spécifiant la situation courante ('action')

output : suite de 10 caractères représentant le nom valide d'un document (cfr. infra)

un nom est valide

si lors d'une correction il existe déjà lors de sa saisie

si lors d'une création il n'existe pas encore lors de sa saisie).

pré : afin de toujours resté dans l'écran, il faut que la coordonnée (x,y) soit telle que

$0 \leq x \leq 49$

$0 \leq y \leq 24$

appelé par : S_LETTRE, S_FORM, S_LET_ST, S_DOC_QCQ, C_DOC, RENOMME, MIS_EN_PAGE

appelle : LIRESTR

fonction :

- lire le nom d'un document

- valider le nom sachant que

un nom est valide

si lors d'une correction il existe déjà lors de sa saisie

si lors d'une création il n'existe pas encore lors de sa saisie).

Module CHOIX TAB.

input : - un tableau de choix possibles (4 valeurs)

- indice majoré du choix courant c-à-d

indice = déplac. + ind. choix

où déplac. = constante égale à 6 soit l'ordonnée sur l'écran du premier choix possible

ind. choix = indice dans le tableau des choix possibles (1 à 4) du choix courant

output : indice du choix courant majoré maj

appelle : REVERSEBLINK, NORMAL

appelé par : C_DOC, CREATION

CHAPITRE IV

fonction :

- affiche le tableau des choix possibles avec le choix courant en vidéo inverse
- permet de changer ce choix courant
- acceptation du choix courant (touche d'acceptation)

Module C DON DOC.

input : données secondaires d'un document

output : données secondaires d'un document maj

appelle : MESSACT, AFF_FCT, FRAME, LIRESTR

appelé par : C_DOC

fonction :

- affiche les valeurs actuelles (ou par défaut) des données du document
- permet de changer ces valeurs
- permet d'accepter une valeur modifiée (touche d'acceptation)
- permet d'accepter l'ensemble des données (touche d'acceptation)

Module ACTION DOC.

input : coordonnées de la ligne où vont avoir lieu les écritures
sur écran
une donnée du document (les 4 mots-clés)
une suite de 15 caractères (mot-clé donné)

output : nom d'un document

appelé par : LIRE_DIR

fonction : affiche sur l'écran le nom d'un document si celui-ci possède un mot-clé égal au mot-clé donné

CHAPITRE IV

3.3.2. MODULES DU GESTIONNAIRE D'ECRAN

Module AFF FCT.

input : code ('choix') représentant l'ensemble des fonctions à afficher sur l'écran

output : écran maj (partie commande)

appelle : AFF_PRES, AFF_CREAT, AFF_COR1, AFF_COR2, AFF_MANIP, AFF_MEP1, AFF_MEP2, AFF_IMPRESS, AFF_TAB, AFF_DICT, AFF_FORM, AFF_DIR, REVERSE, NORMAL

appelé par : S_LETTE, S_FORM, S_LET_ST, S_DOC_QCQ, C_DON_DOC, EFFACE, RENOMME, MEP, ENREG_TAB, UTIL_DICT, CHERCHE, CREATION, CORRECTION, MIS_EN_PAGE, MANIPULATION, IMPRESSION

fonction : affiche en vidéo inverse, dans la partie de commande de l'écran, les fonctions désigné par le code donné.

Module AFF MARGE.

input : abscisse ('md') représentant la position de la marge droite courante abscisse ('pos_d') représentant la position de la nouvelle marge droite

output : écran maj (partie texte)

appelé par : RESTORE_SCREEN, ENREG_TAB

fonction : efface sur l'écran l'ancienne marge située en md et affiche la nouvelle marge spécifiée par l'abscisse pos_d.

Module MESSPOS.

input : code binaire identifiant le type de message

output : écran maj (partie message)

appelé par : ENREGCAR, INSERTCAR, SUPPCAR, REMPLACECAR, EFF_1L, EFF_XL, SOULIGNER (2ème version), HAUT, BAS, GAUCHE, DROITE, PG_DW, PG_UP, SCROLL_DW, SROLL_UP, HOME, G_END, Z_CENTRER, MEP, ENREG_TAB, CHERCHE, JUMPHOLE

fonction : écrit en vidéo inverse, dans la partie message de l'écran, la position courante de telle sorte que
si le type de message = 1, seul l'indice colonne est écrit sur l'ancien
si le type de message = 2, les indices ligne et colonne sont écrits sur les anciens
déplace ensuite le curseur à la position courante.

CHAPITRE IV

Module MESSACT.

output : écran maj (partie message)

appelé par : C_DON_DOC, EFFACE, RENOMME, CREATION, CORRECTION,
MIS_EN_PAGE

fonction : affiche sur la première ligne de l'écran un message
spécifiant l'action courante après avoir effacé tout
autre message sur cette ligne.

Module MESSDEBUT.

input : l'action courante sous forme d'une suite de 21 caractères
un caractère spécifiant le mode du clavier

output : ligne des messages maj

appelé par : ENREG_TAB, MANAGER, PUBLI_POST, NORM_IMPRES

fonction : écrit entièrement la ligne des messages en vidéo
inverse
(cfr. description de cette ligne supra).

Module WRSCREEN.

input : coordonnée ('xax,yax') du point sur l'écran où va se
faire l'écriture
nombre de caractère à écrire ('count')
buffer, carcar

output : écran maj

pré : $1 \leq yax \leq 23$
 $0 \leq xax$
 $0 \leq count$
 $xax + count \leq 79$

appelle : SOULIGNER, REVERSEBLINK, NORMAL

appelé par : EFF_1L

fonction : va à la position (xax,yax) sur l'écran et écrit COUNT
caractères de BUFFER sur l'écran selon le type du
caractère décrit dans CARCAR ; l'écriture se fait à
partir de la position xax dans BUFFER.

CHAPITRE IV

Module RESTORE SCREEN.

input : ordonnée ('pl_ecran') sur l'écran de la première ligne à restaurer
ordonnée ('pl_tab') dans BUFFER de la première ligne à restaurer
nombre de ligne à restaurer ('nb-ligne')
buffer, carcar, position courante,
marge gauche et marge droite

output : écran maj
position courant maj

pré : 2 <= pl_ecran
2 <= pl_tab <= bufmax
0 <= nb_ligne
pl_ecran + nb_ligne <= 23

appelle : AFF_MARGE, SOULIGNER, REVERSEBLINK, NORMAL, BLINK

appelé par : INSERTCAR, SUPPCAR, EFF_1L, EFF_XL, S_LET_ST
PG_DW, PG_UP, SCROLL_DW, SCROLL_UP, Z_CENTRER, MEM,
UTIL_DICT, CHERCHE, JUMPHOLE, CORRECTION

fonction :

- restaure un certain nombre de lignes textes d'un écran c'est-à-dire va à la ligne pl_ecran sur l'écran et écrit nb_ligne+1 lignes textes de BUFFER sur l'écran (tous les caractères sont affichés selon leur type décrit dans CARCAR) à partir de la ligne pl_tab dans BUFFER
- si la dernière ligne texte d'un document est affichée à la ligne l et que l n'est pas la dernière ligne de la partie texte de l'écran courant (ligne 24) alors les lignes l+1 .. 24 de l'écran sont effacées
- la position courante est maj.

Module HAUT.

input : position courante, écran

output : position courante et écran maj

appelle : POS_CR, MESSPOS, SCROLL_DW, HAUT

appelé par : MANAGER, HAUT, GAUCHE

CHAPITRE IV

STRATEGIE :

1. si l est la ligne courante et que l-1 existe alors l-1 est la nouvelle ligne courante
si c est la colonne courante et
 - 2.1. si le nombre de caractères de la ligne l-1 \geq c
alors c reste la colonne courant

l-1	V	O	I	C	I		U	N
l	E	X	E	M	P	L	E	-

l-1	V	O	I	C	I		U	<u>N</u>
l	E	X	E	M	P	L	E	

- 2.2. si le nombre de caractères de la ligne l - 1 $<$ c
alors c est l'indice du dernier caractère de la ligne l - 1

l-1.	U	N						
l	E	X	E	M	P	L	E	-

l-1	U	N	-					
l	E	X	E	M	P	L	E	

2. si elle existe mais n'apparaît pas sur l'écran, alors le contenu de l'écran est déplacé verticalement d'une ligne vers le bas de telle sorte que la ligne l-1 soit la première de l'écran (partie texte) puis retour en 1

3. si elle n'existe pas, le module reste sans effet

fonction : il permet de passer dans un document à la ligne précédente selon la stratégie définie avec maj de la position courante.

CHAPITRE IV

Module BAS.

input : position courante

output : position courante et écran maj

appelle : POS_CR, MESSPOS, SCROLL_UP, BAS

appelé par : MANAGER, BAS, DROITE

stratégie : cfr. module HAUT avec inversion du rôle de l et l-1

fonction : il permet de passer dans un document à la ligne suivante selon la stratégie définie avec maj de la position courante.

Module GAUCHE.

input : position courante

output : position courante et écran maj

appelle : MESSPOS, POS_CR, HAUT

appelé par : MANAGER

stratégie :

soit c la colonne courante

si c-1 existe sur cette ligne (càd c-1 >= marge gauche)
alors c-1 est la nouvelle colonne courante
l reste l'indice de la ligne courante

sinon si l est la ligne courante et que l-1 existe
alors l-1 est la nouvelle ligne courante
c prend la valeur de l'indice du dernier
caractère de la ligne l-1

si l est la ligne courante et que l-1 n'existe pas
alors le module est sans effet

fonction : permet dans un document de passer au caractère précédent selon la stratégie définie avec maj de la position courante.

Module DROITE.

input : position courante
numéro de la dernière ligne
buffer

output : position courante et écran maj

appelle : MESSPOS, BAS

CHAPITRE IV

appelé par :MANAGER

stratégie : soit c la colonne courante

si c+1 existe sur cette ligne (càd c+1 = marge droite)
alors c+1 est la nouvelle colonne courante
l reste l'indice de la ligne courante

sinon si l est la ligne courante et que l+1 existe
alors l+1 est la nouvelle ligne courante
c prend la valeur de l'indice du premier
caractère de la ligne l+1

si l est la ligne courante et que l+1 n'existe pas
alors le module est sans effet

fonction : permet dans un document de passer au caractère suivant
selon la stratégie définie avec maj de la position
courante.

Module PG DW.

input : position courante
numéro de la dernière ligne
l'indicateur de positionnement de l'écran

ouput : écran, position courant maj

appelle : RESTORE_SCREEN, MESSPOS

appelé par : Z_JUSTIFIER, MANAGER

fonction : permet d'afficher l'écran suivant càd 22 nouvelles
lignes du document (ou moins si la fin du document est
atteinte) qui suivent la dernière ligne de l'écran
courant

si la dernière ligne du document n'est pas encore sur
l'écran

- affiche un nouvel écran tel que la dernière ligne
de l'écran courant soit la première du nouvel
écran ; celui-ci est complété par les lignes
suivantes du document
- positionne le curseur dans le coin supérieur droit

si la dernière ligne du document est déjà sur
l'écran courant, le module est sans effet.

Module PG UP.

input : position courante
l'indicateur de positionnement de l'écran

ouput : écran, position courant maj

CHAPITRE IV

appelle : RESTORE_SCREEN, MESSPOS

appelé par : MANAGER

fonction : permet d'afficher l'écran précédent càd 22 nouvelles lignes du document (ou moins si le début du document est atteint) qui précèdent la première ligne de l'écran courant

si la première ligne du document n'est pas encore sur l'écran

- affiche un nouvel écran tel que la première ligne de l'écran courant soit la dernière du nouvel écran ; celui-ci est complété par les lignes précédentes du document
- positionne le curseur dans le coin supérieur droit

si la première ligne du document est déjà sur l'écran courant, le module est sans effet.

Module SCROLL DW.

input : nombre de lignes ('nl') de déplacement
position courante

output : écran et position courante maj

pré : top_tab - nl >= 2

appelle : RESTORE_SCREEN, MESSPOS

appelé par : HAUT, SUPPCAR

fonction :

- fait défiler les lignes de l'écran de telle sorte que
 - . toute ligne est décalée vers le bas du nombre de lignes de déplacement
 - . les lignes inférieures de l'écran (qui débordent) vont disparaître
 - . au dessus de l'écran vont apparaître les lignes précédentes du document (nl lignes)
- maj la position courante.

Module SCROLL UP.

input : nombre de lignes ('nl') de déplacement
position courante

output : écran et position courante maj

pré : top_tab + nl <= bufmax

CHAPITRE IV

appelle : RESTORE_SCREEN, MESSPOS

appelé par : BAS, REMPLACECAR

fonction :

- fait défiler les lignes de l'écran de telle sorte que
 - . toute ligne est décalée vers le haut du nombre de lignes de déplacement
 - . les lignes supérieures de l'écran (qui débordent) vont disparaître
 - . en dessous de l'écran vont apparaître les lignes suivantes du document (nl lignes)
- maj la position courante.

Module HOME.

input : position courante

output : position courante et écran maj

appelle : MESSPOS

appelé par : MANAGER

fonction : déplace le curseur dans le coin supérieur droit de l'écran (partie texte) courant.

Module G END.

input : position courante
indicateur de positionnement de l'écran
numéro de la dernière ligne

output : écran et position courante maj

appelle : MESSPOS

appelé par : MANAGER

fonction : déplace le curseur en fin de l'écran (partie texte) courant càd sur le dernier caractère de la dernière ligne de cet écran.

3.3.3. MODULES DE L'EDITEUR

Module ENREGCAR.

input : un caractère tapé par l'utilisateur CARAC
buffer, carcar, écran
position courante
caractère de fin de ligne
marge droite

output : buffer maj, carcar éventuellement maj,
position courante maj,
écran maj,
caractère de fin de ligne éventuellement maj

post : si (l,c) est la position du dernier caractère
significatif du texte,
alors la position courante
= (l,c+1) si c <= marge droite
= (l+1, marge gauche) sinon.

appelle : WRAP, LIG_SUIV, MESSPOS

appelé par : MANAGER, REFORMAT, SPRING_TAB, UTIL_DICT

fonction : mise dans buffer du caractère tapé par l'utilisateur
déplacement du caractère de fin de ligne
maj de l'écran et de la position courante
maj de carcar si nécessaire.

Module INSERTCAR.

output : un caractère tapé par l'utilisateur CARAC
buffer, carcar, écran
position courante
marge droite et gauche

output : buffer, carcar et écran maj
position courante maj

pré : si (l,c) est la position du dernier caractère significatif
du texte,
la position courante ≠ (l,c+1) si c <= marge droite
≠ (l+1, marge gauche) sinon

appelle : RET_INSERT, POS-CR, SOULIGNER, RESTORE_SCREEN, MESSPOS,
INSERT_DEPLAC

appelé par : MANAGER, L_JUSTIFIER, SPRING_TAB, SUPPCAR

fonction : mise dans buffer du caractère tapé par l'utilisateur
maj de l'écran et de la position courante
maj de carcar si nécessaire

CHAPITRE IV

Module SUPPCAR.

input : buffer, carcar, écran,
position courante, marge gauche,
soulig

output : buffer et carcar maj,
écran maj, position courante maj

appelle: RESTORE_SCREEN, MESSPOS, SCROLL_DW, INSERTCAR,
EFF_REGION, SUPPCAR, POS-CR

appelé par : SUPPCAR, MANAGER

stratégie :

1° cas : le curseur n'est pas placé sur la marge gauche lors
de l'appel à SUPPCAR

mg							md						
E	X	E	M	P	L	E							

mg							md						
E	E	M	P	L	E								

2° cas : le curseur est sur la marge gauche lors de l'appel à
SUPPCAR

mg							md						
							V	O	I	C	I		
U	N		E	X	E	M	P	L	E				

							V	O	I	C	U	N	
E	X	E	M	P	L	E							

CHAPITRE IV

le dernier caractère de la ligne précédente (s'il existe) est effacé et de plus, si un mot (ou plusieurs jusqu'à au maximum la ligne entière) peut être contenu(e) dans la ligne précédente, il(elle) y est transféré(e). La ligne courante peut disparaître entièrement avec réajustement des lignes inférieures.

fonction : efface le caractère se trouvant avant le curseur
maj de buffer, carcar, position courante et de l'écran.

Module REMPLACECAR.

input : une caractère tapé par l'utilisateur CARAC
buffer, carcar, écran, position courante, marge droite

output : buffer, carcar, écran, position courante maj

appelle : SCROLL_UP, MESSPOS

appelé par : MANAGER, SPRINGTAB

fonction : remplace un caractère du texte par le caractère que l'utilisateur a tapé

Module EFF REGION.

input : marq, nb-marq,
buffer, position courante, marge gauche

appelle : COMPARE_MARQ, EFF_1L, EFF_XL, POS-CR

appelé par : MANAGER, SUPPCAR, MOINS_LIGNE

fonction : a pour but d'initialiser les variables de début de zone (t_pc2,pc1) et de fin de zone (i,j).

Module EFF 1L.

input : buffer, carcar, écran
variables de début et de fin de zone
l'indicateur du positionnement de l'écran (top_tab)

output : buffer, carcar, écran, position courante maj
éventuellement indicateur du positionnement de l'écran maj

appelle : WRSCREEN, RESTORE_SCREEN, MESSPOS, POS-CR

appelé par : EFF_REGION

CHAPITRE IV

stratégie :

. d'effacement :

				pc1				j						
t_pc2=i	V	O	I	C	I		U	N		E	S	S	A	I

V	O	I		E	S	S	A	I						
---	---	---	--	---	---	---	---	---	--	--	--	--	--	--

. de maj écran :

- si la zone à effacer est sur l'écran lors de l'exécution de la commande seule cette zone est maj
- sinon, tout l'écran est maj et la ligne contenant la zone à effacer est la première de ce nouvel écran

fonction : efface dans buffer, carcar, et sur l'écran, une zone qui ne s'étend que sur une seule ligne (càd une ligne texte) suivant la stratégie définie ci-dessus.

Module EFF XL.

input : buffer, carcar, écran
variables de début (t_pc2, pc1) et de fin de zone (i, j)
l'indicateur de positionnement de l'écran (top_tab)
marge gauche, numéro de la dernière ligne texte courante

output : buffer, carcar, écran maj
position courante, n° dernière ligne texte maj
éventuellement indicateur du positionnement de l'écran maj

appelle : RESTORE_SCREEN, MESSPOS, POS-CR

appelé par : EFF_REGION

stratégie :

- . de maj de l'écran : cfr. eff_11
- . d'effacement :

CHAPITRE IV

	pc1							
t_pc2	V	O	I	C	␣		U	N
	T	E	X	T	E		D	E
	P	L	U	S	I	E	U	R
i	L	I	G	␣	E	S		A
	E	F	F	A	C	E	R	
	j							

V	O	I	C			
E	S		A			
E	F	F	A	C	E	R

1er étape : efface la fin de la ligne texte où se trouve la marque de début de zone

2eme étape : efface le début de la ligne texte où se trouve la marque fin de zone et justifie cette ligne à gauche

3eme étape : supprime les lignes textes intermédiaires (càd celles situées entre les deux marques - indice t_pc2 et i) s'il y en a et ré-ajuste le texte en déplaçant vers le haut toutes les lignes entre i et la dernière ligne texte

fonction : efface dans buffer; carcar et sur l'écran une zone qui s'étend sur plusieurs lignes textes suivant la stratégie définie ci-dessus

Module SOULIGNER.

1ere version :

positionnement du clavier en mode soulignement.

appelé par : MANAGER, WRAP, INSERTCAR, SUPPCAR, WRSCREEN, RESTORE_SCREEN

CHAPITRE IV

```
fonction : . si le mode est "off" alors positionne le mode a
            "on"
            sinon met celui-ci a "off"

            . indique le (non) positionnement dans le coin
              supérieur droit de l'écran
```

2eme version :

input : buffer, carcar, écran
position courante

```

output : buffer, carcar et écran maj
           position courante maj

```

```
pré : le caractère doit être un caractère significatif
```

appelle : MESSPOS

appelé par : MANAGER

```
fonction : . si le caractère est souligné, génère le même
           caractère non souligné
```

- si le caractère est non souligné, génère le même caractère souligné

Module POSE MARQ.

```

input : indicateur de la phase de traitement ('code')
        indicateur du nombre de caractères marqués ('nb_marq')
        tableau des coordonnées des caractères marqués ('marq')
        indicateur du positionnement de l'écran
        buffer, carcar, pos cour

```

```

output : écran, carcar maj
         tableau des coordonnées des caractères marqués maj
         indicateur du nombre de caractères marqués maj

```

appelle : REVERSEBLINK, NORMAL, BLINK

appelé par : MANAGER

stratégie :

1. Pose des marques de bloc logique

le caractère courant est marqué (code 2 dans carcar) et sa coordonnée est mise dans le tableau des coordonnées ('marq') ; l'écran est maj avec le caractère marqué en vidéo inverse clignotant.

CHAPITRE IV

si avec cette nouvelle marque, le nombre de marque est supérieur à 2 alors, les deux anciennes marques sont effacées.

2. Pose des marques de bloc texte.

le caractère courant est marqué (code 4 dans carcar) ;
l'écran est maj avec le caractère marqué en clignotant.

si le caractère marqué l'était déjà précédemment, il devient caractère non marqué.

fonction : a pour but de désigner un caractère donné comme caractère marqué selon la stratégie définie.

Module ENREG TAB.

input : ligne de présentation courante ('tab_cour')
position courante
caractère tapé par l'utilisateur

output : ligne de présentation maj
marge gauche et droite maj
écran, position courante maj

appelle : AFF_FCT, AFF_MARGE, MESSDEBUT, MESSPOS

appelé par : MANAGER

stratégie :

- pour introduire de nouvelles tabulations, il faut taper la lettre "t" sur un des points de la ligne de présentation entre les lettres "g" et "d"
- pour supprimer une tabulation, il faut taper un "." sur le "t" qui représente la tabulation à supprimer
- pour changer la mg, il faut taper un nouveau "g" sur la ligne de présentation à au moins 30 points à gauche du "d" existant
- pour changer la md, il faut taper un nouveau "d" sur la ligne de présentation à au moins 30 points à droite du "g" existant

fonction : il permet de visualiser et de modifier la ligne de présentation selon la stratégie définie.

Module SPRING TAB.

input : ligne de présentation
buffer

CHAPITRE IV

output : buffer maj

appelle : ENREGCAR, REMPLACECAR, INSERTCAR

appelé par : MANAGER

fonction : à partir de la position courante, enregistre, insère ou remplace par "i" caractères blancs où "i" est la différence entre l'indice de la prochaine tabulation et la colonne courante.

Module LIRE DICT.

input : fichier dictionnaire

output : écran maj

appelé par : UTIL_DICT

fonction : lit le contenu du fichier dictionnaire et l'affiche dans une fenêtre sur l'écran (fenêtre par fenêtre) ; lors de l'affichage, l'identifiant est affiché en vidéo invers et l'utilisateur est invité par un "?" à entrer l'identifiant souhaité.

Module UTIL DICT.

input : fichier dictionnaire
caractères frappés par l'utilisateur

output : écran maj

appelle : AFF_FCT, LIRE_DICT, RESTORE_SCREEN, ENREGCAR

appelé par : MANAGER

fonction :

- saisi l'identifiant et enregistre le texte y correspondant
- si l'utilisateur le désire, affiche le contenu du fichier dictionnaire dans une fenêtre annexe sur l'écran.

Module CHERCHE.

input : suite de caractères à trouver
position courante, buffer,
indicateur de positionnement du tableau

output : message à l'utilisateur
écran maj, position courante maj

appelle : RESTORE_SCREEN, AFF_FCT, MESSPOS

CHAPITRE IV

appelé par : MANAGER

stratégie :

à partir de la position courante

0. faire 1 tant que pas trouvé ou tant que pas fin texte

1. si le mot proposé peut encore se trouver sur la ligne courante

alors comparer le mot proposé avec la suite de caractères de même longueur prise dans BUFFER à partir de la position courante

si égal alors on a trouvé et terminé

sinon on prend dans BUFFER une même suite mais à partir d'une nouvelle position courante égale à l'ancienne + 1 et aller en 1

sinon passer à la ligne suivant et aller en 0

fonction : cherche dans BUFFER, à partir de la position courante, la première occurrence d'une suite de caractères et déplace le curseur au début de cette suite en cas de réussite dans la recherche.

Module JUMPHOLE.

input : buffer, position courant
indicateur de positionnement du tableau

output : position courante, buffer et écran maj
indicateur de positionnement du tableau maj

appelle : RESTORE_SCREEN, MESSPOS

appelé par : MANAGER

fonction : déplace le curseur au début du trou suivant càd sur le premier "." constituant ce trou.

3.3.4. MODULES DU FORMATEUR

Module L CENTRER.

input : numéro de la ligne à centrer ('nl')
 marge droite
 buffer, carcar

output : buffer, carcar, écran maj

appelle : CHK_CR, CALC_BG

appelé par : Z_CENTRER

fonction : permet de centrer une ligne par rapport aux marges
 gauche et droite.

Module L JUSTIFIER.

input : numero de la ligne à justifier ('nl')
 marge droite, marge gauche
 buffer

output : écran maj

appelle : CHK_CR, CALC,BG, INSERTCAR

appelé par : Z_JUSTIFIER

fonction : permet de justifier une ligne par rapport aux marges
 gauche et droite.

Module Z CENTRER.

input : nombre de caractères marqués
 tableau des coordonnées des caractères marqués
 indice de positionnement du tableau

output : nombre de caractères marqués maj
 tableau des coordonnées maj
 indice de positionnement du tableau maj
 écran et position courante maj

appelle : COMPARE_MARQ, L_CENTRER, RESTORE_SCREEN, MESSPOS

appelé par : MANAGER

fonction : permet de centrer une zone de texte càd un certain
 nombre de lignes que l'on a défini par des marques ;

 s'il n'y qu'une seule marque, la ligne comprenant la
 marque est centrée ;

CHAPITRE IV

s'il n'y en a pas, la ligne où se trouve le curseur est centrée.

Module Z JUSTIFIER.

input : numéro de la première ligne à justifier
numéro de la dernière ligne de texte
marge droite, buffer,
indice de positionnement du tableau

output : écran et position courante maj

appelle : L_JUSTIFIER, PG_DW, POS_CR

appelé par : MANAGER

fonction : permet de justifier une zone de texte cad un certain nombre de lignes comprises entre le numéro de ligne donné et la dernière ligne de texte (bornes comprises).

Module MONS LIGNE.

input : buffer
nombre de caractères marqués
numéro de la dernière ligne
marge haut et bas
longueur d'une page

output : buffer et nombre de marques maj

appelle : EFF_REGION

fonction : si le curseur se trouve sur une ligne texte, ce module efface cette ligne ; sinon il est sans effet.

Module INIT VEDETTE.

input : numéro de la ligne de buffer où va se trouver la première ligne de la vedette ('i')
marge droite
renseignements concernant un destinataire ('r_document.entete.dest')

ouput : buffer maj

appelé par : MEP

fonction : copie dans BUFFER, à partir de la coordonnée (i, md-30), les données concernant un destinataire et marque ces lignes de BUFFER comme étant des lignes de présentation.

CHAPITRE IV

Module REFORMAT.

input : fichier texte
 indicateur de soulignement
 buffer

output : buffer et indicateur de soulignement maj

appelle : ENREGCAR

appelé par : MEP

fonction : lit le contenu du fichier texte et enregistre tous les
 caractères dans BUFFER.

CHAPITRE IV

Module TMBM INSERT.

```
input : numéro de la dernière ligne
       indice du début d'une nouvelle page ('deb')
       (= fin de la page précédente)
       marge haut, marge bas
       buffer, carcar
```

```
output :buffer, carcar maj
```

appelé par : MEP

stratégie :

soit $lp = 20$

$$t_m = 1$$
$$bm = 1$$

buffer comme ci dessous :

19	E	T																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
----	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

19	E T
20	
21	- - - - -
22	B O N J O U R
23	L E J O U R

où les lignes 20 et 21 sont deux lignes de présentation.

fonction : insère dans BUFFER les marges haut et bas en fin/début de page selon la stratégie définie.

V. MANUEL D'UTILISATION

Nous décrirons ici les éléments nécessaires (sur la disquette) pour utiliser le traitement de texte, les choses à savoir (dans sa tête) pour l'utiliser et présenterons un exemple d'utilisation du traitement de texte pour créer une lettre et l'imprimer.

Il est important, avant d'aborder ce chapitre, de souligner une hypothèse. Nous supposerons en effet que l'utilisateur a lu et compris les spécifications fonctionnelles du traitement de texte décrites au chapitre II. Sans quoi, ce manuel pourrait avoir très peu d'intérêt.

1. QUE FAUT-IL SAVOIR AVANT L'UTILISATION DU TRAITEMENT DE TEXTE ?

Pour utiliser le traitement de texte, une seule disquette est nécessaire mais elle doit OBLIGATOIREMENT se trouver dans le lecteur de disquette b:.. Tous les documents créés seront stockés sur la disquette se trouvant dans ce lecteur.

La disquette doit contenir les fichiers suivant :

- obligatoirement : TRT.COM
- facultativement : TABULATION
 DICT
 des fichiers d'adresses
 (nom de fichier + '.adr')
- des documents préalablement créés sous la
 forme de deux fichiers :
 - nom de document + '.t'
 - nom de document + '.d'

Si on travaille avec deux disquettes, le fichier TRT.COM peut se trouver sur la disquette rangée dans le lecteur de disquette a:..

2. LE DEMARRAGE DU PROGRAMME

L'entrée dans le programme de traitement de texte se fait en tapant :

- B:TRT si le fichier TRT.COM est dans le lecteur b:
- A:TRT ou TRT si le fichier TRT.COM est dans le lecteur a:..

3. QUE SUFFIT-IL DE SAVOIR PENDANT L'UTILISATION DE TRAITEMENT DE TEXTE ?

3.1. Connaissance de l'écran

- en bas de l'écran apparaissent toutes les actions possibles Les noms sont mnémoniques et significatifs ; toutefois, si on a des problèmes ou des hésitations, on se référera à la liste des actions disponibles (cfr. chap V 4.1).

CHAPITRE V

- en haut de l'écran apparaissent des messages. Si on sait que le message à l'extrême droite "CL I" signifie que le clavier se trouve en mode "insertion" (tout caractère tapé sur le clavier est inséré dans le texte) et que, placé au même endroit, le message "CL R" signifie que le clavier est en mode "remplacement" (tout caractère tapé sur le clavier remplace l'ancien caractère), alors on connaît tout de l'écran.

3.2. Connaissance du clavier

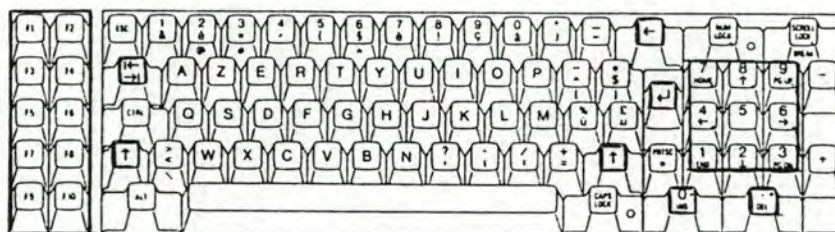


fig. 5.1 : Le clavier de l'Olivetti M-24

- à gauche : il y a les dix touches fonctions notées F1 - F10 dont le but est de sélectionner les actions apparaissant en bas de l'écran.

- à droite :

Si le mode numérique n'est pas sélectionné (pas de voyant rouge à la touche "NUM LOCK", ces touches servent à se déplacer dans le texte ; sinon les touches sont numériques.

Si les graphismes des touches ne paraissent pas assez significatifs, on consultera la liste des commandes de déplacement du curseur (cfr. chap V 4.2 et 4.3).

- touche INS : permet d'alterner les modes "insertion/remplacement" du clavier (cfr. connaissance de l'écran).
- touche DEL : permet d'effacer un caractère (celui sur lequel se trouve le curseur).
- touche ← : permet d'effacer un caractère (celui qui précède le curseur).
- touche ⇐ : permet de déplacer le curseur à la prochaine tabulation.

CHAPITRE V

- touche ↑ : sélection du mode majuscule.
la touche "CAPS LOCK" permet de bloquer ce mode (le voyant lumineux s'allume).
- touche ↵ : caractère de fin de ligne ou d'acceptation de données.

4. EXPLICATION DES TOUCHES ET COMMANDES

4.1. Les touches fonctions

00 ANNUL	10 ANNUL	20 ANNUL	30 ANNUL	40 ANNUL
01 CREAT	11 +ADRES	21 MARQUE	31 MARQUE	41 NORMALE
02 MODIF	12 -ADRES	22 EFF-REG	32 OPTIONS	42 PUBLI-POST
03 MIS PAG		23 CHERCHE	33 MIS PAG	
04 IMPRES	14 +LIGNE	24 +LIGNE	34 +LIGNE	44 ADRESSE
05 MANIP	15 SOULIG	25 SOULIG	35 -LIGNE	45 REPERT
	16 SAUT		36 CENTRER	46 EFFACE
	17 DICT		37 JUSTIFIER	47 RENOMME
	18 TAB			48 REC DOC
	19 ENREG	29 ENREG	39 ENREG	

Les commandes ainsi implémentées ont l'objectif suivant :

Annul (00, 10, 20, 30, 40) : annule l'opération en cours
c'est à dire l'action courante qui est mentionnée dans la
ligne des messages (cfr.description écran).

Creat (01) : création d'un document.

Modif (02) : modification d'un document.

Mis pag (03, 33) : mise en page d'un document.

Impres (04) : impression d'un document.

Manip (05) : manipulation d'un document.

+Adres (11) : permet d'ajouter une adresse à une liste
existante ou non.

-Adres (12) : permet d'éliminer une adresse d'une liste.

+ligne (14, 24, 34) : créer une fenêtre d'une ligne
blanche dans le texte en dessous de la position du
curseur.

CHAPITRE V

Soulig (15, 25) : positionne le mode "soulignement" on/off.

Saut (16) : permet le déplacement du curseur de trou en trou dans un formulaire.

Dict (17) : permet d'utiliser le dictionnaire pour l'introduction rapide de texte.

Tab (18) : permet de modifier les tabulations et les marges.

Enreg (19, 2, 3) : permet d'enregistrer un document sur le support externe.

Marque (21, 31) : indique qu'un caractère du texte est une marque d'un bloc logique ou d'un bloc texte.

Eff - reg (22) : permet d'effacer une région c'est-à-dire une suite de caractères comprise entre deux caractères marqués.

Cherche (23) : recherche dans le texte une occurrence d'une chaîne de caractères donnée à partir de la position courante.

Options (32) : permet de faire apparaître le formulaire avec les options de formatage pour éventuellement les modifier.

- LIGNE (35) : permet de supprimer une ligne texte.

Centrer (36) : permet de centrer une ligne texte.

Justifier (37) : permet de justifier toutes les pages d'un document.

Normale (41) : permet de commander une impression d'un document.

Publi-post (42) : permet de commander une impression de plusieurs documents (cfr. def. publi- postage).

Adresse (44) : permet la gestion des listes d'adresses.

Repert (45) : permet de faire défiler sur l'écran le répertoire.

Efface (46) : permet de supprimer un document du support externe.

Rennome (47) : permet de changer le nom d'un document.

Rec doc (48) : permet de rechercher la liste des documents qui possèdent un mot clé donné.

CHAPITRE V

4.2. Les commandes

en plus des 10 touches claviers (F1-F10), il y a aussi :

- 8 touches fonctions programmées (clavier numérique en mode non numérique)

flèche à gauche	: recule d'un caractère
flèche à droite	: avance d'un caractère
flèche en haut	: monte d'une ligne sur l'écran
flèche en bas	: descend d'une ligne sur l'écran
home	: va au début d'un écran
END	: va à la fin d'un écran
PG UP	: va à l'écran précédent
PG DW	: va à l'écran suivant

- 3 fonctions particulières

BS (backspace) : efface le caractère précédent la position courante

Del : efface le caractère à la position courante

Ins : positionne les modes insertion/remplacement

- 1 touche de tabulation

(cfr. fig. 3.2)

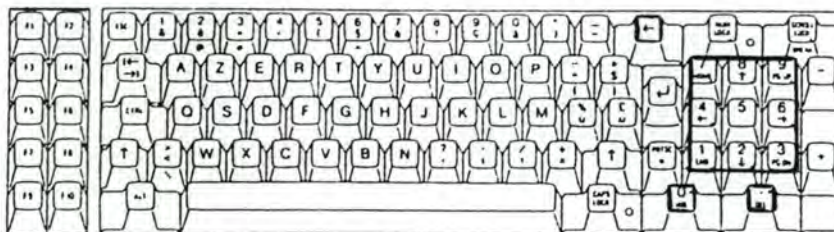


fig.5.2 : Les touches spéciales du clavier

CHAPITRE V

5. UN EXEMPLE D'UTILISATION

Supposons que l'on veuille créer une lettre ordinaire et en demander l'impression.

On dispose du fichier programme (TRT.COM) sur la disquette située dans le lecteur b: et on se trouve au niveau commande c'est-à-dire que "A 3" apparaît sur mon écran.

On tape B:TRT ; l'écran suivant apparaît :

TRAITEMENT DE TEXTE							
0	1créat	2modif	3mis pag	4impres	5manip		0annul

On choisit la fonction "créat" pour la création du document en enfonçant la touche F1 ; l'écran suivant apparaît :

CREATION DE DOCUMENT	
lettre lettre standard formulaire document quelconque	
0 0annul

CHAPITRE V

Le programme propose, initialement, de créer une lettre (l'option "lettre" est affichée en vidéo inverse) ; si on est d'accord avec l'écran proposé, on accepte en tapant ; l'écran suivant apparaît dès lors :

LETTRE		
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p>Nom de la lettre :</p> <p>Sujet</p> <p style="padding-left: 20px;">mot clé 1 :</p> <p>Expéditeur :</p> <p>Date d'émission : 16/8/1985</p> <p>Destinataire</p> <p style="padding-left: 20px;">Nom :</p> <p style="padding-left: 20px;">Qual :</p> <p style="padding-left: 20px;">Adr,n° :</p> <p style="padding-left: 20px;">Cp,Loc :</p> </div>		
0	 0annul

On remplit une à une les données demandées ; chacune d'elles se termine par la touche d'acceptation.

Seul le nom de la lettre est OBLIGATOIRE.

Remplissons toutes les données mais pour le destinataire, ne donnons que son nom. Dans ce cas, le système cherche dans ses fichiers d'adresses s'il existe un destinataire ayant ce nom. Si c'est le cas, les autres données sont complétées automatiquement (syst.) ; l'écran suivant apparaît :

LETTRE		
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p>Nom de la lettre : DOC1.....</p> <p>Sujet</p> <p style="padding-left: 20px;">mot clé 1 : ESSAI.....</p> <p>Expéditeur : EXPET.....</p> <p>Date d'émission : 16/8/1985</p> <p>Destinataire</p> <p style="padding-left: 20px;">Nom : DUPONT.....</p> <p style="padding-left: 20px;">Qual : administrateur..... (syst.)</p> <p style="padding-left: 20px;">Adr,n° : rue Blondeau 54.... (syst.)</p> <p style="padding-left: 20px;">Cp,Loc : 5000 Namur..... (syst.)</p> </div>		
0	 0annul

CHAPITRE V

Dès que l'on a accepté tous les renseignements, l'écran suivant apparaît pour la création de la lettre proprement dite :

CREATION DOCUMENT /col: 1/lig: 1 doc1 CL I							
1							
1	4+ ligne	5soulig	6	7dict	8tab	9enreg	0annul

On introduit le texte de la lettre suivant :

"Vos beaux yeux, Marquise, me font mourir d'amour".

On a donc l'écran ci après :

CREATION DOCUMENT /col:1 /lig: 1 doc1 CL I							
Vos beaux yeux, Marquise, me font mourir d'amour							
1							
1	4+ ligne	5soulig	6	7dict	8tab	9enreg	0annul

CHAPITRE V

Afin de mémoriser le texte de la lettre ainsi que toutes les données y correspondant, on enfonce la touche F9.

On retrouve alors le premier écran de la séquence.

TRAITEMENT DE TEXTE						
0	1créat	2modif	3mis pag	4impres	5manip	0annul

Afin de faire imprimer la lettre que l'on vient de mémoriser sur la disquette, on va maintenant sélectionner l'option 4 pour demander une impression.

L'écran suivant va apparaître :

IMPRESSION DOCUMENT			
3	1normale	2publi p	0annul

CHAPITRE V

On sélectionne l'option 5 pour demander une impression normale d'un document.

On obtient alors l'écran suivant :

IMPRESSION DOCUMENT		
<div>Nom du document :</div>		
3		0annul

Dès que le nom du document est donné, l'écran est complété comme suit :

IMPRESSION DOCUMENT		
<div>Nom du document : DOC1.....</div> <div>Vérifiez que l'imprimante bien branchée, s.v.p. ?</div>		
3		0annul

CHAPITRE V

Après vérification, il suffit d'enfoncer la touche d'acceptation pour demander l'impression de la lettre "DOC1" ; mais il est toujours possible de l'annuler en choisissant l'option d'annulation F10.

CONCLUSIONS

A la suite de la conception, de l'implémentation et de la programmation d'un traitement de texte qui se voulait essentiellement original au niveau de son interface et de sa facilité d'utilisation, il nous semble utile de faire une évaluation du travail réalisé dans le cadre de ce mémoire.

Au niveau de la conception, on peut se demander jusqu'où les objectifs généraux ont été réalisés.

Il est très difficile de répondre à cette question car il est impossible de prouver que les choix qui ont été faits pour faciliter l'utilisation sont optimaux. De même, il est difficile de justifier les choix qui ont été faits au niveau de l'interface. La seule évaluation possible serait de tester comparativement notre traitement de texte avec d'autres. Mais là encore, ce n'est pas une chose facile car il y aura un problème dans l'échantillonnage des personnes choisies pour faire ces tests. En effet, il faudrait que la même personne fasse le test de plusieurs traitements de texte et en plus, que l'influence d'un programme testé soit nulle vis-à-vis des autres tests.

Cependant, la voie est ouverte et notre traitement de texte est près et ne demande qu'à être évalué.

Au niveau de l'implémentation, nous retiendrons deux remarques.

La première est une nouvelle voie ouverte pour une amélioration possible, si toutefois, le besoin s'en fait sentir. Il s'agit d'implémenter la fonction de vidage du tableau dans des fichiers de travail afin de pouvoir produire des documents excédant 350 pages.

La seconde est une critique de la manière dont la gestion d'écran a été implémentée. Nous avons réalisé une gestion d'écran par procédures. On peut se demander si la gestion n'aurait pas été plus concise et plus adaptée si elle avait été réalisée à l'aide d'un gérant d'écran basé sur des fichiers descripteurs.

Au niveau de la programmation et de sa validité, nous retiendrons la chose suivante.

La construction du programme a eu lieu en cinq grandes étapes, distinctes. Chacune d'elle a débuté par le développement des modules de plus bas niveau (bottom-up). Ceux-ci ont été testés de façon à passer au moins une fois par chaque branche de l'algorithme correspondant. A l'étape suivant, nous avons créé les modules de niveau supérieur en utilisant les procédures déjà existantes, la méthode de test restant la même. Nous avons tenté d'analyser toutes les situations critiques possibles ; cependant, il est clair que nous ne pouvons pas affirmer que le traitement de texte fonctionne dans tous les cas.

Pour clôturer cette évaluation, précisons quelque peu ce que nous retirons personnellement de ce mémoire. Le fait d'être confrontés à un problème informatique relativement complet a été très enrichissant. Il s'agit certainement de la partie la plus personnelle de nos études et celle qui se rapproche le mieux d'un travail d'informaticien. Une fois de plus, nous nous sommes rendus compte qu'il est essentiel de distinguer conception et implémentation, et que, dans ce contexte, certains concepts sont parfois difficiles à situer.

LEXIQUE

Ce petit lexique est destiné à définir certains termes ou expressions spécifiques introduits dans ce travail.

On entendra dès lors :

1. CARACTERE DE FIN DE LIGNE.

C'est le dernier caractère d'une ligne texte. C'est un caractère spécial, non visible pour l'utilisateur, mais qui est explicitement introduit par l'utilisateur lorsqu'il tape le caractère de retour du chariot, ou implicitement par le système si la marge droite est atteinte.

2. CARACTERE SIGNIFICATIF.

C'est tout caractère se trouvant dans une ligne texte à l'exception du caractère de fin de ligne.

3. METTRE UNE SUITE DE CARACTERES A BLANC.

C'est une manière d'initialiser une suite de caractères à laquelle on ne veut pas donner de valeur spécifique ; la suite peut être considérée soit comme une suite vide ou soit comme une suite ne comprenant que des caractères blancs (code ASCII 32 - blank space).

4. TOUCHE D'ACCEPTATION.

Dans la phase de saisie de données, on parlera souvent de touche d'acceptation : il s'agit de la touche (code ASCII 13). Cette touche est en fait la traditionnelle touche de retour de chariot que l'on retrouve sur les machines à écrire et que l'on retrouvera dans la phase de traitement du texte avec cette fonctionnalité de "retour à la ligne suivante".

Comme son nom l'indique, elle sert notamment pour :

- accepter des données quand elles sont proposées par le système (ex : une date)
- accepter des données quand elles sont entrées par l'utilisateur (toute donnée demandée par le système doit se terminer par cette touche)
- accepter une grille de données telle qu'elle apparaît sur l'écran

5. DONNEES D'UN DOCUMENT.

A ce sujet, on distinguera parfois les données intrinsèques et les données secondaires qui se différencient essentiellement au niveau de la dynamique de leur utilisation (fréquence de maj, d'utilisation, ...).

Par données intrinsèques, il faut entendre :

- nom d'un document
- état d'un document
- date d'un document

Par données secondaires, il faut entendre :

- expéditeur du document
- destinataire du document
- mots-clés

6. RE-AJUSTEMENT D'UN TEXTE.

Il s'agit simplement d'une concaténation verticale d'un texte qui contiendrait des blocs logiques non-contigus ; par exemple, un texte contient des blocs non-contigus après la suppression d'un bloc logique au sein du texte.

7. TROU DANS UN FORMULAIRE.

Dans un formulaire, on appelle un trou toute suite de plus d'un point ('.') consécutif dans le texte. Cette suite est en fait la représentation de l'espace disponible jugé nécessaire pour remplir ce formulaire.

exemple : NOM :
 PRENOM :

le formulaire ci-dessus comprend deux trous.

BIBLIOGRAPHIE

- (1) MARTINEAU, J. (1982),
La bureautique,
Edition Mac Graw-Hill
- (2) MEYROWITZ, N. and VAN DAM A. (September 1982),
"Interactive Editing Systems : Part I and II",
Computing Surveys, Vol. 14, N° 3, 321 - 415
- (3) AIESEC Namur (avril 1985)
Microfac - Traitement de texte
- (4) STANDARD OIL COMPAGNY - Indiana (july 1980),
"Office Technology Systems Conceptual Functionnal
Requirements from a user's perspective"
Office Technology Department
- (5) VIRGATCHICK, I. (1984),
"Guide Marabout du traitement de texte",
Collection Marabout Service
- (6) LOCASYST,
Manuel d'opération,
Traitement de texte Wordstar
- (7) VAN LOO, R. (1984),
"Guide Marabout du Macintosh",
Collection Marabout Service
- (8) I.B.M.,
Manuel d'opération,
Traitement de texte Profesional Editor
- (9) Michael HAMMER, Richard ILSON, Tim ANDERSON, Edwart GILBERT,
Michael GOOD, Bahram NIAMIR, Larry ROSENSTEIN, and Sandor
SCHAICHET (1981)
"The implementation of ETUDE,
An Integrated and Interactive Document Production System"
- (10) W. HORAK, G. KRÖNERT
"An object-oriented office document architecture model for
processing and interchange of documents"
Second ACM - SIGOA Conference on Office Information Systems
(June 25-27, 1984 Toronto Canada
Edited by Clarence A. Ellis Vol. 5, N° 1-2) ACM

Ouvrages généraux

Hal GLATZER,
Introduction au traitement de texte,
Sybex

"Le traitement de texte"
A. Huberman et J.C. Meneau
Informatique et entreprise Eyrolles

Modes d'emploi de traitement de texte

"Mode d'emploi de Apple Writer 2.0",
Notes par E. DUBOIS,
du Club de micro-informatique du Collège St Michel à Bruxelles

"Alfaword",
Notes de présentation du traitement de texte,
de l'Institut technico-commercial de Libramont

Ce document a été édité et formaté à l'aide
du traitement de texte

VOLKSWRITER DE LUXE